

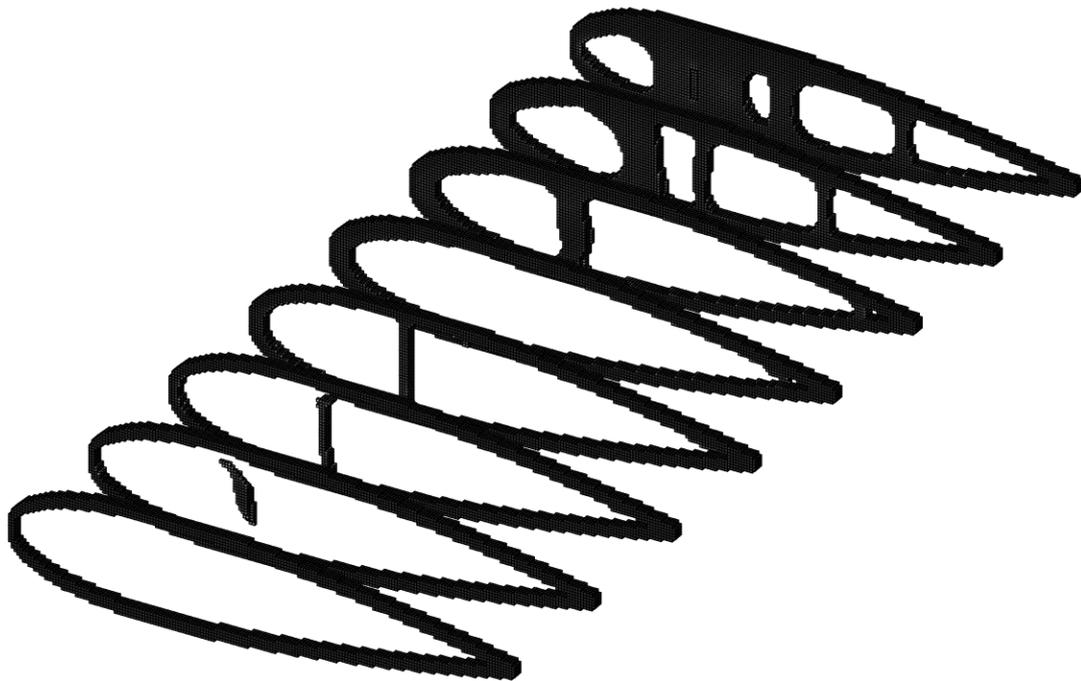
MASTER OF SCIENCE THESIS

Topology optimization using the Finite Cell Method

Exploiting multi-resolution p-FEM to increase computational efficiency of topology optimization applications

J.P. Groen B.Sc.

19-08-2015



Faculty of Aerospace Engineering · Delft University of Technology
Faculty of Mechanical Engineering · Technical University of Denmark



Topology optimization using the Finite Cell Method

Exploiting multi-resolution p-FEM to increase computational efficiency of topology optimization applications

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace Engineering
at Delft University of Technology

&

For obtaining the degree of Master of Science in Wind Energy at
Technical University of Denmark

J.P. Groen B.Sc.

19-08-2015

Faculty of Aerospace Engineering · Delft University of Technology
Faculty of Mechanical Engineering · Technical University of Denmark



Copyright © J.P. Groen B.Sc.
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
AEROSPACE STRUCTURES AND COMPUTATIONAL MECHANICS

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled “**Topology optimization using the Finite Cell Method**” by **J.P. Groen B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 19-08-2015

Chairman of the exam committee:

dr. C. Kassapoglou

Supervisor TU Delft:

dr. M. Ruess

Supervisor TU Delft:

dr. M. Langelaar

Supervisor DTU:

prof. dr. O. Sigmund

Reader:

prof. dr. G.J.W. van Bussel

Summary

The ongoing demand for better performing designs, has resulted in an increase in the complexity of topology optimization problems. Traditionally, the majority of the corresponding computational cost comes from solving the analysis equations using linear finite elements (FE). In this thesis a topology optimization method is presented, that is based on the finite cell method (FCM). This higher-order fictitious domain method is, due to its decoupled geometry-, integration-, and analysis-mesh well suited for large-scale topology optimization, and reducing its corresponding computational cost.

A topology optimization model using the FCM is developed, which performs density based topology optimization, and includes the well-known SIMP method for material interpolation. Gradient based optimization is performed using the method of moving asymptotes (MMA). The use of the decoupled density and analysis mesh, requires regularization using mesh-independent filtering. Furthermore, this decoupling allows a wide variety in analysis systems, ranging from very simple to highly accurate analysis. Different values for the polynomial degree, and the corresponding amount of density voxels per cell have been tested, to identify for which settings topology optimization could be performed in a reliable and accurate fashion.

It is observed that the use of a decoupled density and analysis mesh greatly reduced the computational cost of topology optimization compared to linear FEM. Especially in 3D topology optimization examples, the computational has been decreased by more than a factor 10, while maintaining a high-resolution in the density field. The use of a larger length-scale can reduce the computational cost even more, which is especially beneficial for robust topology optimization.

For the relatively simple optimization examples used in this thesis, no setting could be identified as being the best, regarding accuracy and computational cost. The choice of the analysis system completely depends on the complexity of the optimization problem. The simple optimization problems showed great increase in computational efficiency using relatively low polynomial degree $p = 1, 2, 3$, hence a benefit for higher-polynomial degrees was not identified. For more difficult topology optimization examples, such as stress-constrained topology optimization, or examples with multiple weak boundary conditions, a more accurate analysis system is required, hence a larger polynomial degree should be used.

Acknowledgements

This thesis concludes my time as a master student in the European Wind Energy Master. Obviously, I could not have done this with the help of my supervisors Ole Sigmund, Martin Ruess and Matthijs Langelaar. They enormously improved my knowledge on solid mechanics and topology optimization, and guided my research in the right direction.

Second of all, I want to thank Nils Zander who's program FCMLab formed the basis of my optimization model. Furthermore, I would like to thank Krister Svanberg, who allowed me to use the MMA algorithm in my optimization program.

I want to thank my fellow graduates Paul Thedends and Paul Salden with whom I had very interesting discussions on topology optimization. Thanks to Michiel Wassenaar and Lourens Blok for reading the drafts of my thesis and pointing me at my spelling mistakes.

But most importantly, I want to thank all my friends, my family, and my girlfriend who made my time as a student such a great time.

Delft, The Netherlands
19-08-2015

J.P. Groen B.Sc.

Contents

Summary	v
Acknowledgements	vii
List of Figures	xii
List of Tables	xv
Nomenclature	xvii
1 Introduction	1
1.1 Research objective	2
1.2 Context of research	2
1.2.1 Topology optimization	2
1.2.2 Advancements in analysis methods	3
1.2.3 New developments within topology optimization	4
1.3 Thesis structure	4
2 Methodology	5
2.1 The finite cell method	5
2.1.1 From variational formulation to the finite element method	5
2.1.2 Higher-order shape functions	7
2.1.3 The fictitious domain approach	9
2.1.4 Enforcement of boundary conditions in the weak formulation	10
2.2 Topology optimization	12
2.2.1 Topology optimization using the SIMP method	12
2.2.2 Gradient based optimization	13
2.2.3 Controlling the quality of the solution	15
2.3 FCMTOP: the Finite Cell Method Topology Optimization Program	18
2.3.1 Capabilities of FCMTOP	18
2.3.2 Optimization problems	19
2.3.3 The optimization process	21

3	The validity of topology optimization using the FCM	23
3.1	The limits of FCMTOP	23
3.2	Stable parameters for minimum compliance problems	26
3.3	Stable parameters for minimum displacement problems	30
3.4	Concluding remarks on the accuracy, and the stability of FCMTOP	35
4	Efficient topology optimization using the FCM	37
4.1	Drivers of computational cost in topology optimization	37
4.1.1	The computational cost of one design iteration	37
4.1.2	The size and shape of the stiffness matrix	38
4.1.3	The effect of p and n_{voxel} on computational cost	40
4.1.4	Concluding remarks on the computational cost of one design iteration	42
4.2	Experiments on computational efficiency	42
4.2.1	Michell truss	43
4.2.2	3D MBB-Beam example	45
4.3	Concluding remarks	46
5	Discussion on the potential of topology optimization using the FCM	49
5.1	The performance of topology optimization using the FCM	49
5.1.1	The reliability, efficiency and accuracy experiments	49
5.1.2	Topology optimization examples with a larger length-scale	50
5.1.3	The use of higher-order shape functions	51
5.2	Large-scale topology optimization using FCMTOP	53
6	Final remarks	55
6.1	Conclusions	55
6.2	Recommendations	56
	References	57
A	A literature review on topology optimization methods	65
A.1	Applications of structural optimization	65
A.2	Problem formulation in topology optimization	66
A.3	Topology optimization using a continuous density distribution	67
A.4	Topology optimization using the level-set method	70
A.5	Topology optimization using the phase field approach	70
A.6	Topology optimization using discrete evolutionary approaches	71
A.7	Topology optimization for large scale optimization problems	72
A.7.1	Reduction of model complexity	72
A.7.2	Reduction of the FE-analysis time	73
A.7.3	Increase in optimization efficiency	73
A.7.4	Parallel topology optimization	74
B	Minimum compliance experiments	75
C	Minimum displacement experiments	87

List of Figures

1.1	Example of topology optimization on a MBB beam, the top figure denotes the design domain Ω with the applied loads and boundary conditions, and the bottom figure shows the corresponding optimized structure [60]	1
2.1	"Set of one-dimensional standard and hierarchic shape function for $p = 1, 2, 3$ " [30]	7
2.2	2-Dimensional mode types [85]	8
2.3	"The fictitious domain approach: the physical domain Ω_{phys} is extended by the fictitious domain Ω_{fict} into an embedding domain Ω to allow easy meshing of complex geometries. The influence of Ω_{fict} is penalized by α " [58]	9
2.4	Artificially stiff checkerboard patterns on a MBB-beam optimization example	15
2.5	Design domain an boundary conditions for the MBB-beam problem	19
2.6	Design domain an boundary conditions for the compliant force inverter problem	20
2.7	Schematic overview of the optimization process	21
3.1	Artificially stiff 'optimized' MBB-Beam, no filter is used, $n_{voxel} = 5$, $p = 3$, and $V_{max} = 0.4$	24
3.2	Split domain experiment to show the error in the displacement field, when a low polynomial degree is used	24
3.3	The displacement in x -direction at the location of the void	24
3.4	The displacement in x -direction at the location of the three voxel wide strip of void $n_{voxel} = 5$, and $p = 3$	25
3.5	The MBB-Beam example for two different settings, both values of the compliance C are obtained with $n_{voxel} = 1$, and $P = 3$. A discretization of 360 by 120 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	26
3.6	The MBB-Beam example, $n_{voxel} = 5$, $p = 2$, $\hat{F}(\rho) = 1.11MJ$, and $C = 1.20MJ$. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	27
3.7	The MBB-Beam example for different settings. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths .	27
3.8	The MBB-Beam example for two different filter settings, with $n_{voxel} = 4$, and $P = 2$. A discretization of 180 by 60 voxels is used, and $r_{min} = 2$ voxel widths	30

3.9	The compliant inverter example for three settings. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	31
3.10	The compliant inverter for two settings where an instability at the beginning of the optimization causes a connection between the input spring and the weak boundary. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	32
3.11	The compliant inverter for two settings where the hinges are either connected by one node, or are completely disconnected. A discretization of 120 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths	34
3.12	The compliant inverter for two settings where the connection between the weak boundary and input spring negatively influenced the objective. A discretization of 120 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths	35
4.1	Overview of the non-zero indices of stiffness matrix \mathbf{K} , when a polynomial degree sorting scheme is used. The matrix corresponds to the MBB-beam example, where a discretization of 180 by 60 voxels is used	40
4.2	Overview of the non-zero indices of stiffness matrix \mathbf{K} , when the reverse Cuthill-McKee algorithm is used to optimize the bandwidth. The matrix corresponds to the MBB-beam example, where a discretization of 180 by 60 voxels is used	41
4.3	Indication of the variations in T_{chol} , and $T_{assembly}$ for every design iteration, in the 3D MBB-beam example	43
4.4	The Michell truss topology optimization example, problem formulation, and optimized design	43
4.5	The optimized density distribution for the 3D MBB-beam optimization example, with $p = 2$ and $n_{voxel} = 4$. A discretization of 120 by 40 by 40 voxels is used, and sensitivity filtering is applied	47
5.1	The shape of a NACA 0020 airfoil, and the corresponding pressure distribution obtained in XFOIL using a viscous analysis is performed at sea-level conditions ($\rho_{air} = 1.225 \text{ kg/m}^3$, $T = 291.25 \text{ K}$), $Mach = 0.2$, and ($\alpha = 5^\circ$)	54
5.2	Different cross-sections of the clamped wing example. A discretization of 200 by 40 by 400 voxels is used, density filtering with modified Heaviside projection is applied, with $r_{min} = 2$ voxel widths	54
A.1	Example of topology optimization on a MBB beam, the top figure denotes the design domain Ω with the applied loads and boundary conditions, and the bottom figure shows the corresponding optimized structure [60]	66
A.2	This figure shows the effect of the different penalization factors q on the density of the material, it can be seen that a high penalty factor makes intermediate densities uneconomical	68
A.3	Solution for a MBB beam to demonstrate the occurrence of the checkerboard problem.	69
A.4	"Exampplles of different types of geometry mapping, (a) Conforming: Only the material domain is discretized. (b) Immersed Boundary Technique: Boundary conditions are enforced locally at the interface. (c) Density-based: In each finite element an element density indicates the amount of material" [74]	70
A.5	"Examples of the phase field function: (a) A 2D Example represented by the phase field function, (b) A 1D illustration of the phase field function" [73]	71
A.6	Decoupling of the displacement mesh, and design variable mesh using MTOP. (a) Displacement mesh, (b) Superposed meshes, (c) Design Variable mesh [47]	73
A.7	"Amdahl's law for parallel speedup" [2]	74

List of Tables

2.1	Overview of the capabilities of FCMTOP	18
3.1	Results of the stability experiments using sensitivity filtering, the compliance (C) is shown	28
3.2	Results of the stability experiments using density filtering, the compliance (C) is shown	28
3.3	Results of the stability experiments using density filtering with Heaviside projection, the compliance (C) is shown	29
3.4	Results of the stability experiments using density filtering with modified Heaviside projection, the compliance (C) is shown	29
3.5	Results of the stability experiments using robust topology optimization ($\eta = 0.2$), the compliance (C) is shown	29
3.6	Results of the stability experiments using sensitivity filtering, the values for the output displacement u_{out} are shown	32
3.7	Results of the stability experiments using density filtering, the values for the output displacement u_{out} are shown	33
3.8	Results of the stability experiments using density filtering with Heaviside projection, the values for the output displacement u_{out} are shown	33
3.9	Results of the stability experiments using density filtering with modified Heaviside projection, the values for the output displacement u_{out} are shown	33
3.10	Results of the stability experiments using robust topology optimization ($\eta = 0.2$), the values for the output displacement u_{out} are shown	34
3.11	The amount of iterations per optimization of the MBB-beam for different settings. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	36
4.1	The size of the stiffness matrix (n_{dof}), for different polynomial degrees (p), and different cell topologies. A discretization of 120 by 120 voxels is used	39
4.2	The number of non-zero elements (n_{nz}), for different polynomial degrees (p), and different cell topologies. A discretization of 120 by 120 voxels is used	39
4.3	The required memory to store the stiffness matrix, for different polynomial degrees (p) and different cell topologies. A discretization of 120 by 120 voxels is used	40

4.4	The setting that will be used for the numerical experiments on efficiency are indicated by the green cells, the settings marked purple are computationally too expensive, while the red cells correspond to unstable settings	42
4.5	The results on numerical cost for different p , and n_{voxel} , using the Michell truss optimization example. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied	44
4.6	The results on numerical cost for different p , and n_{voxel} , using the Michell truss optimization example, including bandwidth optimization. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied	45
4.7	The results on numerical cost for different p , and n_{voxel} , using the 3D MBB-beam optimization example. A discretization of 120 by 40 by 40 voxels is used, and sensitivity filtering is applied	46
5.1	The results on numerical cost for different p , and n_{voxel} , using the compliant force inverter example. A discretization of 240 by 120 voxels is used, robust topology optimization is applied ($\eta = 0.2$), with $r_{min} = 4$ voxel widths	50
5.2	The objective and post-processed compliance, for different p , and n_{voxel} , using the Michell truss optimization example. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied	52
B.1	Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	76
B.2	Results of the stability experiments on the MBB-beam example. A discretization of 240 by 80 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	77
B.3	Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths	78
B.4	Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths	79
B.5	Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths	80
B.6	Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths	81
B.7	Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths	82
B.8	Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths	83
B.9	Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths	84
B.10	Results of the stability experiments on the MBB-beam example. A discretization of 240 by 80 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths	85
C.1	Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	88

C.2	Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths	89
C.3	Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths	90
C.4	Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths	91
C.5	Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths	92
C.6	Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths	93
C.7	Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths	94
C.8	Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths	95
C.9	Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths	96
C.10	Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths	97

Nomenclature

Latin Symbols

a_i	Coefficients corresponding to the MMA
\mathbf{B}	Strain displacement matrix
\mathbf{b}	Body force tensor
C	Compliance
c_i	Coefficients corresponding to the MMA
\mathbf{C}	Elasticity tensor
\mathbf{D}	Unknown coefficients
E	Young's modulus
\hat{F}	Objective function
\mathbf{F}	Load vector
\hat{G}	Constraint function
H_{ei}	Weighting function for filter, indicates distance between elements e and i
\mathbf{K}	Stiffness matrix
k	Iteration index
L_j	Legendre polynomial of the j^{th} order
m	Number of constraint functions
M_{nd}	Measure of non-discreteness
\mathbf{N}	Shape function matrix
n_{bw}	Half-bandwidth of the stiffness matrix
n_c	Number of cells
n_{dof}	Number of degrees of freedom in the stiffness matrix
n_e	Number of elements/voxels
n_{iter}	Number of iterations

n_{nz}	Number of non-zero elements
n_{sc}	Total number of voxels/sub-cells per cell
n_{voxel}	Number of voxels in each direction of a cell
p	Polynomial degree
q	Penalization factor
$T_{assembly}$	Time required to assemble the stiffness matrix
T_{chol}	Time required to perform Cholesky decomposition
T_{opt}	Time required to perform one design iteration
\mathbf{t}	Traction
\mathbf{u}	Displacement field
V	Volume
\mathbf{x}	Location in global coordinate system (x, y, z)

Greek Symbols

α	Material indicator
β	Heaviside smoothness parameter
τ_N	Stabilization parameter in Nitsche based boundary condition
τ_P	Penalization parameter in boundary condition using penalty approach
Γ_D	Dirichlet boundary condition
Γ_N	Neumann boundary condition
ε	Strain tensor
ζ	Move-limit
η	Threshold value
λ	Lagrange multipliers
ν	Adjoint vector
ν	Poisson's ratio
ξ	Location in element coordinate system (ξ, η, ζ)
Π_c	Constraint potential
Π_{ext}	Potential energy acting on domain Ω
Π_{in}	Internal potential energy of domain Ω
Π_{tot}	Total potential energy of domain Ω
ρ	Material distribution
$\boldsymbol{\rho}$	Design vector
$\boldsymbol{\sigma}$	Stress tensor
ϕ	Integrated Legendre polynomial
Ω	Design domain

Abbreviations

BC	Boundary Conditions
CAD	Computer Aided Design
FCMTOP	Finite Cell Method Topology Optimization Program
FCM	Finite Cell Method
FEM	Finite Element Method
MEMS	Micro Electro-Mechanical Systems
MMA	Method of Moving Asymptotes
MTOP	Multi-resolution Topology OPTimization
NURBS	Non-Uniform Rational B-Splines
SIMP	Solid Isotropic Material with Penalization
VWP	Virtual Work Principle

Chapter 1

Introduction

In all disciplines of engineering there is the ever-increasing desire for better performing designs. In structural engineering this is usually translated into a demand for stiffer and lighter structures, hence the increased use of composite materials in automotive and aerospace applications. Apart from new materials and production processes, the computational analysis tools have advanced rapidly. Nowadays, structural engineers rely heavily on these tools, of which the finite element method (FEM) is the most renowned.

The increasing availability of computational power, and maturation of analysis methods has also initiated a new design field called: structural optimization. The core idea is to translate design requirements into a constrained optimization problem, which is solved by a computer program. Structural optimization encompasses three different methods, of which topology optimization is the most fundamental. It dictates the location and connectivity of material in a design domain [23]. Topology optimization therefore seems the ideal structural design. Given a design domain and boundary conditions, a computer program will come up with the ideal design as can be seen in Figure 1.1.

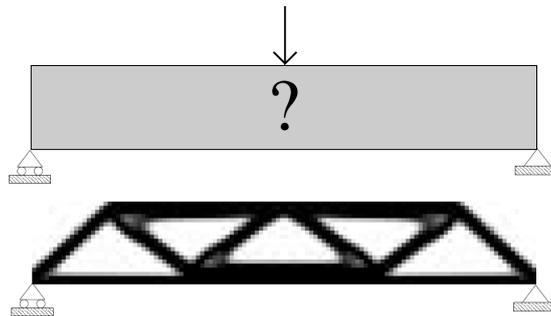


Figure 1.1: Example of topology optimization on a MBB beam, the top figure denotes the design domain Ω with the applied loads and boundary conditions, and the bottom figure shows the corresponding optimized structure [60]

The demand for better performing designs, also has resulted in a dramatic increase in the complexity of topology optimization problems. Large-scale topology optimization requires

enormous computational effort, and is only possible using parallel computation [2]. For topology optimization to become a fully mature design tool, more complex load conditions, boundary conditions, and buckling criteria have to be incorporated, as well [40]. Despite the large amount of studies performed on improving topology optimization, the vast majority of methods still makes use lower-order finite elements [53], which are known to be less accurate compared to higher-order shape functions [13, 71].

In the research reported in this thesis, the effect of higher-order shape functions for topology optimization applications was investigated using the finite cell method (FCM) introduced by Parvızian, Düster and Rank [31, 52]. This higher-order fictitious domain method is, due to its decoupled geometry-, integration-, and analysis-mesh well suited for topology optimization.

1.1 Research objective

The research objective was: *"To investigate the potential of the FCM for large scale, high resolution topology optimization problems, compared to the use of finite elements (FE) with linear shape functions in terms of reliability, efficiency, and accuracy."* Hence, the goal was to develop a method to perform topology optimization with the FCM and identify in which cases it can be more beneficial to use FCM than classical FEM.

To reach this objective, a topology optimization model based on the FCM had to be derived, and validated. This validation process was an extremely difficult task, since topology optimization problems are generally non-convex, but, it defines the accuracy and reliability of the developed model. The most feasible way to tackle the validation process was to compare the optimized structures to well known examples, such as the MBB-Beam or the force-inverter. At the end of this process the parameters were identified at which the developed model is able to perform topology optimization in a stable fashion.

Afterwards, the developed model was compared to a topology optimization model using linear FEM for different optimization experiments. The goal of these experiments was to identify differences in efficiency between both methods and to determine which methods are the most effective for which conditions. Furthermore, the optimization model using the FCM is able to enforce boundary conditions in the weak sense, such that more interesting optimization examples have been used.

1.2 Context of research

In this section a brief introduction will be given on the history and applications of topology optimization, and, the advancement in analysis methods will be discussed. Finally the research reported in this thesis is put into context with respect to the current studies on topology optimization.

1.2.1 Topology optimization

The introduction of topology optimization by Bendsøe and Kikuchi marked the starting point of a rapidly growing and very active research field [15, 27, 64]. Within a couple of years the field divided into a number of different subfields. The research reported in this thesis is focused on the most mature and most used method: the continuous density method using

the solid isotropic material with penalization (SIMP) method for the material interpolation. The reader who is interested in a more detailed literature study on topology optimization should turn to Appendix A.

In the SIMP method the design domain is split into small elements, the density of which can vary between 0 and 1, where 0 corresponds to an element containing no material (void), and where 1 corresponds to an element consisting of material (solid). The density affects the element stiffness in a similar fashion to that of fictitious domain methods. To steer the element densities towards a crisp solid-void solution, the intermediate densities are penalized using the SIMP method. This method which was independently developed by Bendsøe, and Zhou and Rozvany is sometimes referred to as power-law approach [14, 87].

Despite the fact that topology optimization is a relatively new technique it already found applications in various industries such as the aerospace and automotive industries [86, 90]. Furthermore, the method has given very promising results for the biomedical industry and the development of micro electro-mechanical systems (MEMS) [17].

1.2.2 Advancements in analysis methods

Since its introduction in the late 1950's, the FEM has been the most used method for structural analysis. Originally, this method made use of lower-order shape functions, however, the p-version of the finite element method introduced by Babuška, Szabó, and Katz shows superior convergence towards the exact solution [13]. In FE-analysis the physical domain has to be discretized into a mesh. Advancements in meshing and powerful mesh generators allow for efficient meshing of most domains, but, for highly complex geometries the meshing process can still prove to be a time consuming process [58]. To cope with this problem Hughes et al. introduced isogeometric analysis [25]. Here the same functions that are used to represent the geometry in computer aided design (CAD) tools are used to describe the solution field. The FCM goes one step further and is also able to omit the time consuming meshing process even when no CAD model is available.

The FCM is a fictitious domain method, where the embedding domain consists of a physical and fictitious part. Since the boundaries of the physical domain do not necessarily conform with the element boundaries, boundary conditions (BCs) have to be applied in the weak sense. The use of higher-order shape functions and a composed integration scheme make this method very efficient, especially for image based-models such as human femur analysis [57, 58, 83].

Topology optimization can be seen as a fictitious domain method. Therefore the FCM shows great potential for topology optimization examples. The higher-order shape functions and the composed integration scheme have demonstrated the FCM's advantage over linear FEM, and the application of boundaries in the weak sense allows very interesting optimization examples.

In nested topology optimization efficient assembly of the stiffness matrix is required. Therefore, all elements generally have the exact same properties, such that the integration only has to be performed once. The cells in the FCM can all have the same dimensions and shape functions, such that the same benefit can be exploited. Finally, it has to be noted that the developers of the FCM have implemented a heuristic optimization method to show the topology optimization possibilities using the FCM [53].

1.2.3 New developments within topology optimization

Despite the large availability of computational facilities, topology optimization still comes with high computational cost. Research on reducing this cost can be split into three aspects: reducing the complexity of the model, reducing the FE-analysis time, and making the optimization process more efficient.

Model complexity involves the number of degrees of freedom in the stiffness matrix. To reduce this number Stainko proposed a method where the mesh is adaptively refined along the interface between solid and void [66]. Another, more interesting method to reduce the complexity of the model is multi-resolution topology optimization (MTO) described by Nguyen et al [47]. In a more recent paper an adaptive-MTO method is introduced in which the design variable and density fields can be adjusted and refined [48]. Similar to the FCM, a composed integration scheme is used to map the density distribution on the finite element mesh. The multi-resolution methods only include linear shape functions, and imposes a large length-scale, hence, the full potential of the increased resolution is not yet employed.

Several researchers have focused on reducing the analysis time. They have performed studies on recycling parts of the previous stiffness matrix [80], and on approximations of the displacement vector [7, 8]. These advanced methods can be implemented together with the FCM.

The overall optimization time can be decreased by using parallel computations. Borvall and Petersson were the first to describe topology optimization using parallel computing techniques [18]. To utilize the computational facilities to the fullest, the code has to be completely parallelized [2, 1]. The FCM can be parallelized, but, that was beyond the scope of this thesis work [58].

1.3 Thesis structure

A very important part of the reported research was to develop a topology optimization program using the FCM. To do this the Finite Cell Method Topology Optimization Program (FCMTOP) was created. The corresponding theory will be discussed in Chapter 2. First, the FCM used to calculate the displacement field will be introduced, second, the theory behind topology optimization using the SIMP method will be discussed. Chapter 2 is concluded with an overview of how the program works.

The limitations of the model will be demonstrated in Chapter 3, where the focus will be on the accuracy and the stability of the developed method. The research into the efficiency of the developed topology optimization method will be discussed in Chapter 4. Theoretical aspects regarding efficient matrix storage and Cholesky decomposition will be discussed, furthermore, the performance of the model will be compared to a topology optimization model using linear FEM. The performance of the developed topology optimization program using the FCM, will be discussed in chapter 5. This thesis is concluded in Chapter 6. The work performed is summed up and the most important results will be emphasized. Recommendations regarding the use of the method in future studies are given.

Chapter 2

Methodology

The development of the topology optimization method based on the finite cell method (FCM) will be discussed in this chapter. First the FCM, used to obtain the displacement field is introduced, second, the theory behind topology optimization is described. Both methods will be combined into the finite cell method topology optimization program (FCMTOP), which will be discussed in the last section.

2.1 The finite cell method

The FCM can be seen as an extension of the classical finite element method (FEM). The FEM will be introduced in Section 2.1.1, subsequently, higher-order shape functions will be discussed in Section 2.1.2. The theoretical background of the FCM and its integration scheme will be shown in Section 2.1.3. This section will be concluded by a discussion on the enforcement of the boundary conditions in the weak sense.

2.1.1 From variational formulation to the finite element method

Consider a deformable body with domain Ω , at which a body force \mathbf{b} is acting. The boundary of the domain consists of a Neumann boundary Γ_N , a Dirichlet boundary Γ_D , and a free boundary. The total potential energy Π_{tot} of this domain is at a stationary point,

$$\Pi_{tot} = \Pi_{in} + \Pi_{ext} \quad (2.1)$$

The foundation of the finite element equations is the virtual work principle (VWP), with which this stationary point can be found [45]. The VWP states that for any kinematic admissible displacement field $\delta\mathbf{u}$ the change in the total potential energy should be equal to zero:

$$\delta\Pi_{tot} = \delta\Pi_{in}(\mathbf{u}, \delta\mathbf{u}) + \delta\Pi_{ext}(\delta\mathbf{u}) = 0 \quad \forall \mathbf{x} \in \Omega, \forall \delta\mathbf{u} \in \mathbf{u} \quad (2.2)$$

where $\delta\Pi_{in}$ and $\delta\Pi_{ext}$ denote the internal and external work on Ω . For linear elastic behavior,

$$\delta\Pi_{in}(\mathbf{u}, \delta\mathbf{u}) = - \int_{\Omega} \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} dV \quad \forall \mathbf{x} \in \Omega \quad (2.3)$$

$$\delta\Pi_{ext}(\delta\mathbf{u}) = \int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} dV + \int_{\Gamma_N} \mathbf{t} \cdot \delta\mathbf{u} d\Gamma_N \quad \forall \mathbf{x} \in \Omega \quad (2.4)$$

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad \forall \mathbf{x} \in \Omega \quad (2.5)$$

$$\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] \quad \forall \mathbf{x} \in \Omega \quad (2.6)$$

$$\mathbf{t} = \mathbf{t}_N \quad \forall \mathbf{x} \in \Gamma_N \quad (2.7)$$

$$\mathbf{u} = \mathbf{u}_D \quad \forall \mathbf{x} \in \Gamma_D \quad (2.8)$$

where $\boldsymbol{\sigma}$ denotes the stress tensor, subsequently $\boldsymbol{\varepsilon}$ denotes the strain tensor, \mathbf{C} is the elasticity tensor, \mathbf{t}_N is the prescribed traction, and \mathbf{u}_D is the prescribed displacement.

Substituting all expressions in Equation 2.2 gives:

$$- \int_{\Omega} \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} dV + \int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} dV + \int_{\Gamma_N} \mathbf{t}_N \cdot \delta\mathbf{u} d\Gamma_N = 0 \quad (2.9)$$

Using the Bubnov-Galerkin method this continuum equation can be discretized into small elements for which the unknown displacement \mathbf{u} can be approximated [24]. Using shape functions \mathbf{N}_e and unknown coefficients \mathbf{d}_e , the displacement at any point \mathbf{x} within the element of interest e can be obtained,

$$\mathbf{u}(\mathbf{x}) = \mathbf{N}_e(\boldsymbol{\xi}(\mathbf{x}))\mathbf{d}_e \quad (2.10)$$

where the global position vector $\mathbf{x} = (x, y, z)$ is mapped on the element coordinate system $\boldsymbol{\xi} = (\xi, \eta, \zeta)$. Since the displacement gradient is symmetric there exists a relation between the strain variation and the variation in unknown coefficients:

$$\delta\boldsymbol{\varepsilon} = \mathbf{B}_e \delta\mathbf{d}_e \quad (2.11)$$

where \mathbf{B}_e is called the strain displacement matrix,

$$\mathbf{B}_e = \nabla\mathbf{N}_e \quad (2.12)$$

Discretization of Equation 2.9 yields,

$$\begin{aligned} - \sum_e \int_{\Omega_e} \delta\mathbf{d}_e^T \mathbf{B}_e^T \mathbf{C} \mathbf{B}_e \mathbf{d}_e dV_e + \sum_e \int_{\Omega_e} \delta\mathbf{d}_e^T \mathbf{N}_e^T \mathbf{b} dV_e \\ + \sum_e \int_{\Gamma_{N,e}} \delta\mathbf{d}_e^T \mathbf{N}_e^T \mathbf{t}_{N,e} d\Gamma_{N,e} = 0 \end{aligned} \quad (2.13)$$

For now, the enforcement of both the boundary conditions (BCs) is not considered. These conditions will be discussed in detail in Section 2.1.4. Furthermore, the element unknowns \mathbf{d}_e are constant and can therefore be taken out of the integral. Equation 2.12 has to hold for any kinematic admissible variation in unknowns and can therefore be written as:

$$- \sum_e \int_{\Omega_e} \mathbf{B}_e^T \mathbf{C} \mathbf{B}_e dV_e \mathbf{D} + \sum_e \int_{\Omega_e} \mathbf{N}_e^T \mathbf{b} dV_e = 0 \quad (2.14)$$

where \mathbf{D} is the vector with containing all unknowns. This problem can be written into the finite element equation as:

$$\mathbf{K}\mathbf{D} = \mathbf{F} \quad (2.15)$$

where \mathbf{K} is the global stiffness matrix assembled from the element stiffness matrices \mathbf{k}_e , while load vector \mathbf{F} can be assembled from the element load vector \mathbf{f}_e .

$$\mathbf{k}_e = \int_{\Omega_e} \mathbf{B}_e^T \mathbf{C} \mathbf{B}_e dV_e \quad (2.16)$$

$$\mathbf{f}_e = \int_{\Omega_e} \mathbf{N}_e^T \mathbf{b} dV_e \quad (2.17)$$

2.1.2 Higher-order shape functions

The shape functions required to interpolate the displacement are not unique, i.e. different sets of basis functions can be used. In classical FEM low-order Lagrange polynomials were used as shape functions, where the unknown coefficients \mathbf{D} , corresponded to the nodal displacements [72]. These lower order shape functions cannot resemble complex displacement fields within an element. To cope with this two options exist: either more smaller elements can be used, or more complex basis functions have to be utilized.

The first option is called the h-version of FEM. In this method the error in strain energy converges linearly with increasing degrees of freedom [24]. In the latter method, the p-version of the finite element method, the element size is kept constant and the polynomial degree is increased. Compared to the linear convergence of the h-version of FEM, the p-version shows exponential convergence towards the exact solution for increasing degrees of freedom [13, 71].

The FCM includes the p-version of FEM. The higher order basis functions used in this method are integrated Legendre polynomials. Contrary to Lagrange polynomials, Legendre polynomials are hierarchic, i.e. the shape functions for polynomial degree p are also included when degree $p+1$ is used as can be seen in Figure 2.1. This property results in a more empty stiffness matrix, allowing an efficient solution of the analysis equations.

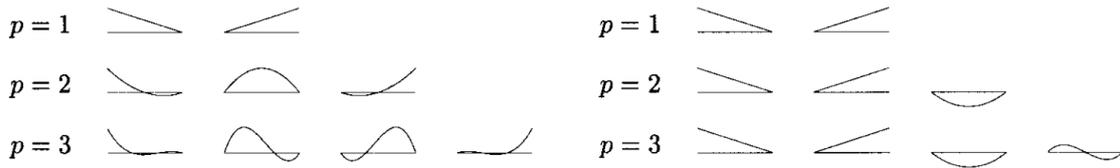


Figure 2.1: "Set of one-dimensional standard and hierarchic shape function for $p = 1, 2, 3$ " [30]

The one dimensional set of shape functions can be defined as:

$$\begin{aligned} N_1(\xi) &= \frac{1}{2}(1 - \xi) \\ N_2(\xi) &= \frac{1}{2}(1 + \xi) \\ N_i(\xi) &= \phi_{i-1}(\xi), \quad i = 3, 4, \dots, p + 1 \end{aligned} \quad (2.18)$$

where ϕ corresponds to the integrated Legendre polynomial,

$$\phi_j(\xi) = \sqrt{\frac{2j-1}{2}} \int_{-1}^{\xi} L_{j-1}(t) dt = \frac{1}{\sqrt{4j-2}} L_j(\xi) - L_{j-2}(\xi), \quad j = 2, 3, \dots \quad (2.19)$$

where $L_j(\xi)$ are the Legendre polynomials. The first two shape functions are the nodal modes corresponding to FEM using linear shape functions. With the integrated Legendre polynomials as basis functions, the displacement can be interpolated:

$$\mathbf{u}(\xi) = N_1(\xi)d_1 + N_2(\xi)d_2 + \sum_{i=3}^{p+1} N_i(\xi)d_i \quad (2.20)$$

The orthogonality property of the Legendre polynomials can be demonstrated for the shape functions where $i \geq 3$:

$$\int_{-1}^1 \frac{dN_i}{d\xi} \frac{dN_j}{d\xi} d\xi = \delta_{ij}, \quad i \geq 3 \text{ and } j \geq 1, \text{ or } i \geq 1 \text{ and } j \geq 3 \quad (2.21)$$

The shape functions can be used in two- or three-dimensional problems, by combining two lower order bases using the tensor product [85]:

$$\begin{aligned} N_{i,j}^{2D}(\xi, \eta) &= N_i^{1D}(\xi)N_j^{1D}(\eta) \\ N_{i,j,k}^{3D}(\xi, \eta, \zeta) &= N_{i,j}^{2D}(\xi, \eta)N_k^{1D}(\zeta) \end{aligned} \quad (2.22)$$

On a standard quadrilateral elements, three different types of modes can be distinguished shown in Figure 2.2 [85].

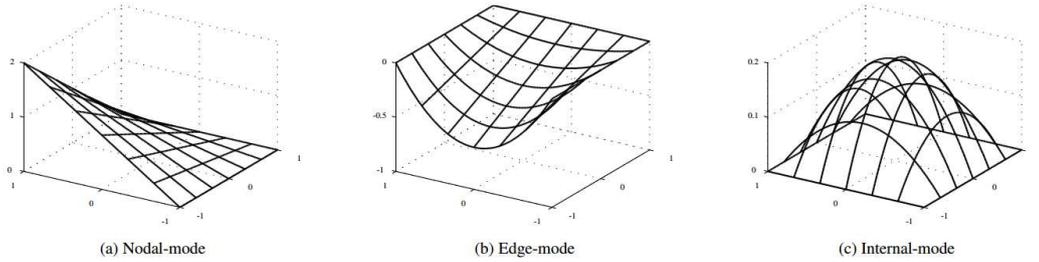


Figure 2.2: 2-Dimensional mode types [85]

- **Nodal Modes**

These standard bi-linear shape functions make sure that the shape function is exactly 1, at the corresponding edge node, and zero at the other edge node, the corresponding mode can be written as,

$$N_1^{n1}(\xi, \eta) = \frac{1}{4}(1 - \eta)(1 - \xi) \quad (2.23)$$

- **Edge Modes**

These modes have a finite value along the corresponding edge and are zero at all other edges. The corresponding mode can be written as,

$$N_i^{e1}(\xi, \eta) = \frac{1}{2}(1 - \eta)\phi_i(\xi) \quad \text{for } i \geq 2 \quad (2.24)$$

- **Internal Modes**

This mode purely gives finite values at the interior of the element, and is zero at all edges. The corresponding mode can be written as,

$$N_{i,j}^{int}(\xi, \eta) = \phi_i(\xi)\phi_j(\eta) \quad \text{for } i, j \geq 2 \quad (2.25)$$

Next to the full tensor product basis, there is another basis called the trunk space. In this basis the internal behavior is described using less shape functions, resulting in a limited number of unknowns [71, 72]. Finally, it should be noted that the FCM can be extended to use other higher-order approximation bases, such as B-splines and non-uniform rational B-splines (NURBS) [58].

2.1.3 The fictitious domain approach

Similar to density based topology optimization methods, the FCM is a fictitious domain approach. This means that the embedding domain Ω consists of a physical domain of interest Ω_{phys} and a fictitious domain Ω_{fict} . A graphical illustration of this method can be seen in Figure 2.3 [58].

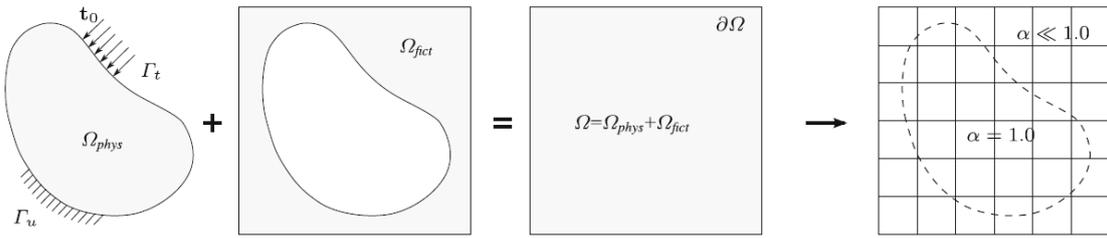


Figure 2.3: "The fictitious domain approach: the physical domain Ω_{phys} is extended by the fictitious domain Ω_{fict} into an embedding domain Ω to allow easy meshing of complex geometries. The influence of Ω_{fict} is penalized by α " [58]

The main advantage of the FCM is that the embedded domain can be discretized into axis-aligned rectangular cells. Since the boundary of the physical domain can cut a cell, the boundary conditions have to be applied in the weak sense, as will be discussed in Section 2.1.4. The analysis is performed on the solution mesh consisting of the cells, while a separate integration mesh and the higher order basis functions will allow for an accurate solution [58].

To take the fictitious domain into account the constitutive matrix as well as the body force vector are multiplied with a material indicator α :

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_{phys} \\ 0 & \forall \mathbf{x} \in \Omega_{fict} \end{cases} \quad (2.26)$$

$$\mathbf{k}_c = \int_{\Omega_c} \mathbf{B}_c^T \alpha \mathbf{C} \mathbf{B}_c dV_c \quad (2.27)$$

$$\mathbf{f}_c = \int_{\Omega_c} \mathbf{N}_c^T \alpha \mathbf{b} dV_c \quad (2.28)$$

To avoid ill conditioning α is chosen to be not exactly 0 but rather 10^{-10} [31, 52].

In the FCM a composed integration scheme is applied to cope with the non-uniform material distribution within a cell. Each cell is divided into rectangular sub-cells with constant material properties. The stiffness matrix and load vector are integrated in the sub-cells and then mapped on the cells. Since both cells and sub-cells are rectangular, the geometry can be interpolated using only linear shape functions, resulting in a constant Jacobian matrix [58, 31].

$$\mathbf{k}_c = \sum_{i=1}^{n_{sc}} \mathbf{k}_i \quad \mathbf{f}_c = \sum_{i=1}^{n_{sc}} \mathbf{f}_i \quad (2.29)$$

A benefit of this integration method is that it is well-suited for voxel-based models, such as qCT-scan data or density distributions used in topology optimization [58, 83]. Each of the voxels e can be represented by a sub-cell, containing $(p + 1)^3$ integration points. The simplicity of the FCM is that each cell has exactly the same dimensions and amount of sub-cells. Therefore, the amount of voxels n_e can be calculated as:

$$n_e = n_c n_{sc} \quad (2.30)$$

where n_c , and n_{sc} correspond to the amount of cells, and sub-cells respectively. Topology optimization application makes use voxel-based models, so in the remainder of this thesis the word voxel is used instead of the word sub-cell. The topology of a cell is therefore defined by the amount of voxels it contains (n_{sc}), which depends on the number of voxels in each direction of a cell (n_{voxel}),

$$n_{sc} = n_{voxel}^{dim} \quad (2.31)$$

where dim corresponds to the dimension of the problem. Hence, each cell has the same amount of voxels in each direction. In case of isotropic material, with a constant Poisson's ratio ν , and a voxel dependent Young's modulus E_i the elasticity matrix can be written as:

$$\mathbf{C}_i = E_i \mathbf{C}^0 \quad (2.32)$$

where \mathbf{C}^0 is independent of the voxel. The cell stiffness can then be defined as follows:

$$\mathbf{k}_c = \sum_{i=1}^{n_{sc}} \alpha_i E_i \mathbf{k}_i^0 \quad (2.33)$$

The voxel matrices \mathbf{k}_i^0 are independent of the voxel-wise material properties, and all cells have the same size and voxel distribution, therefore, the integrands \mathbf{k}_i^0 have to be pre-computed for just one cell. Afterwards, the cell stiffness matrix can be assembled efficiently by multiplying the precomputed matrices with the material constants. These cell matrices can then be mapped into the global stiffness matrix \mathbf{K} .

2.1.4 Enforcement of boundary conditions in the weak formulation

In the FCM the complicated meshing process is shifted to a more complex enforcement of the BCs. Since the BCs not necessarily have to coincide with the cell edges, they have to be applied in the weak sense.

Weak enforcement of Neumann boundary conditions

Homogeneous Neumann BCs are always satisfied in the FCM. The stiffness penalization in the fictitious domain makes sure that the stresses cannot be transferred beyond Ω_{phys} [52, 31].

Non-zero Neumann BCs can be simply enforced by integrating over the traction boundary. This only requires an explicit representation of the load surface or an approximation based on small parameterized surface elements. For now consider a parameter dependent surface description (Γ_t), with parametric coordinates u and v [58]. The influence of the traction load on the load vector can be written as:

$$\mathbf{F}^t = \sum_e \int_{\Gamma_{t,e}} \mathbf{N}_e^T \mathbf{t}_e \|\mathbf{x}_{,u} \times \mathbf{x}_{,v}\| d\Gamma_{t,e} \quad (2.34)$$

where $\mathbf{x}(u, v)$ is the location on the parameterized surface, and $\|\mathbf{x}_{,u} \times \mathbf{x}_{,v}\|$ is the vector normal to the surface plane at point (u, v) .

Weak enforcement of Dirichlet boundary conditions

For the application of the essential BCs the potential energy is augmented with a constraint potential Π_c , resulting in a new potential:

$$\tilde{\Pi}_{tot} = \Pi_{in} + \Pi_{ext} + \Pi_c \quad (2.35)$$

Three methods will be considered for the application of Dirichlet BCs in the weak sense. These are the penalty method, the method using identified Lagrange multipliers and Nitsche's method. This is because Nitsche's method, which has been used for the research in this thesis, can be seen as a combination of the first two methods. Just as in the case of the Neumann BCs all methods require an explicit boundary representation on which:

$$\mathbf{u} - \mathbf{u}_D = \mathbf{0} \quad \forall \mathbf{x} \in \Gamma_D \quad (2.36)$$

Using the penalty method the constraint potential can be written as:

$$\Pi_c = \int_{\Gamma_D} \frac{1}{2} \tau_P (\mathbf{u} - \mathbf{u}_D)^2 d\Gamma_D \quad (2.37)$$

where τ_P is the penalization value. For the augmented total potential energy ($\tilde{\Pi}_{tot}$) a stationary point has to be found. The variation in the constraint potential can be written as:

$$\delta\Pi_c = \int_{\Gamma_D} \tau_P (\mathbf{u} - \mathbf{u}_D) \cdot \delta\mathbf{u} d\Gamma_D \quad (2.38)$$

The stationary point can be found by finding the variation of the augmented total potential energy and setting it equal to zero:

$$- \int_{\Omega} \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} dV + \int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} dV + \int_{\Gamma_N} \mathbf{t}_N \cdot \delta\mathbf{u} d\Gamma_N + \tau_P \int_{\Gamma_D} \mathbf{u} \cdot \delta\mathbf{u} d\Gamma_D - \tau_P \int_{\Gamma_D} \mathbf{u}_D \cdot \delta\mathbf{u} d\Gamma_D = 0 \quad (2.39)$$

In case identified Lagrange multipliers are used, the constraint potential can be written as:

$$\Pi_c = \int_{\Gamma_D} \boldsymbol{\lambda} \cdot (\mathbf{u} - \mathbf{u}_D) d\Gamma_D \quad (2.40)$$

where $\boldsymbol{\lambda}$ corresponds to the Lagrange multipliers. The Lagrange multipliers can be identified as:

$$\boldsymbol{\lambda} = -\boldsymbol{\sigma} \cdot \mathbf{n} \quad (2.41)$$

where \mathbf{n} is the normal pointing out of the surface. The variation of the constraint potential can then be defined as follows:

$$\delta\Pi_c = - \int_{\Gamma_D} \delta\boldsymbol{\sigma} \cdot \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_D) d\Gamma_D - \int_{\Gamma_D} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \delta\mathbf{u} d\Gamma_D \quad (2.42)$$

The variation in the augmented total potential energy becomes:

$$- \int_{\Omega} \delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma} dV + \int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} dV + \int_{\Gamma_N} \mathbf{t}_N \cdot \delta\mathbf{u} d\Gamma_N - \int_{\Gamma_D} \delta\boldsymbol{\sigma} \cdot \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_D) d\Gamma_D - \int_{\Gamma_D} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \delta\mathbf{u} d\Gamma_D = 0 \quad (2.43)$$

The final method that will be considered is developed by Nitsche [49]. This method can be seen as an extension of the identified Lagrange multiplier method using a stabilization term

similar to the penalization method. The corresponding constraint potential can be written as:

$$\Pi_c = \int_{\Gamma_D} -\boldsymbol{\sigma} \cdot \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_D) d\Gamma_D + \int_{\Gamma_D} \frac{1}{2} \tau_N (\mathbf{u} - \mathbf{u}_D)^2 d\Gamma_D \quad (2.44)$$

where τ_N is the stabilization parameter. Numerical experiments have shown that a value in the range of 100 to 1000 gives an accurate boundary representation. The corresponding variation of the constraint potential then becomes:

$$\delta \Pi_c = - \int_{\Gamma_D} \delta \boldsymbol{\sigma} \cdot \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_D) d\Gamma_D - \int_{\Gamma_D} \boldsymbol{\sigma} \cdot \mathbf{n} \cdot \delta \mathbf{u} d\Gamma_D + \int_{\Gamma_D} \tau_N (\mathbf{u} - \mathbf{u}_D) \cdot \delta \mathbf{u} d\Gamma_D \quad (2.45)$$

which can be substituted in the total variation of potential energy and be solved for the stationary point.

2.2 Topology optimization

In this section topology optimization using the solid isotropic material with penalization (SIMP) method is discussed. This starts with the introduction of the topology optimization problem in Section 2.2.1, subsequently, the optimization algorithm and the method to obtain design sensitivities will be discussed. Finally, the need to control the quality of the solution will be demonstrated, including the methods to do so.

2.2.1 Topology optimization using the SIMP method

Topology optimization can be seen as a material distribution problem. The goal is to find that optimum material distribution $\rho(\mathbf{x})$ that minimizes an objective function \hat{F} . This function is subject to m constraints \hat{G}_i , of which the first is generally a volume constraint. The mathematical form of the topology optimization problem can be written as [64]:

$$\begin{aligned} \min_{\rho} : \hat{F}(\rho) = F(\rho, \mathbf{u}(\rho)) &= \int_{\Omega} f(\rho, \mathbf{u}(\rho)) dV \\ \text{s.t.} : \hat{G}_1(\rho) = \int_{\Omega} \rho(\mathbf{x}) dV - V_{max} &\leq 0 \\ : \hat{G}_i(\rho) = G_i(\rho, \mathbf{u}(\rho)) &\leq 0, & i = 2, \dots, m \\ : 0 \leq \rho(\mathbf{x}) \leq 1, & & \forall \mathbf{x} \in \Omega \end{aligned} \quad (2.46)$$

where f is a local function, e.g. the strain energy density for minimum compliance problems. The material distribution is allowed to vary between 0 and 1 for gradient based optimization, and \mathbf{u} is solved using the FCM in a separate step. This optimization problem is called a nested topology optimization problem, since the total potential energy is at a stationary point for each optimization step.

To solve Equation A.1, the design domain Ω is discretized into voxels, to which a density is assigned ρ_e . All voxel densities together form the design vector $\boldsymbol{\rho}$. The discretized optimization

problem can be written as:

$$\begin{aligned}
\min_{\boldsymbol{\rho}} : & \hat{F}(\boldsymbol{\rho}) = F(\boldsymbol{\rho}, \mathbf{D}) \\
\text{s.t.} : & \hat{G}_1(\boldsymbol{\rho}) = \mathbf{v}^T \boldsymbol{\rho} - V_{max} \leq 0 \\
& : \hat{G}_i(\boldsymbol{\rho}) = G_i(\boldsymbol{\rho}, \mathbf{D}) \leq 0, \quad i = 2, \dots, m \\
& : \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1} \quad \forall \mathbf{x} \in \Omega
\end{aligned} \tag{2.47}$$

Where \mathbf{v} is the vector containing the element volumes, and V_{max} is the maximum allowed volume of the material in the design domain. A schematic way to solve this nested topology optimization problem is:

1. Create an initial domain $\boldsymbol{\rho}^{(0)}$, where the iteration index $k = 0$.
2. For a given iteration point $\boldsymbol{\rho}^{(k)}$, calculate $\hat{F}(\boldsymbol{\rho}^{(k)})$, $\hat{G}_i(\boldsymbol{\rho}^{(k)})$, and their corresponding sensitivities for $i = 1, \dots, m$.
3. Using an optimization algorithm to find the next iteration point $\boldsymbol{\rho}^{(k+1)}$.
4. Check whether convergence criteria are met, and if not go back to step 2.

Even though the density distribution is allowed to vary between 0 and 1, crisp solid-void structures are generally desired. This can be obtained by penalizing the the stiffness of voxels with intermediate densities using the SIMP method [61].

$$E_e = E_{min} + \rho_e^q (E - E_{min}) \tag{2.48}$$

where q is the penalization factor, where E is the stiffness of a solid voxel, and where E_{min} is a very small value ($\sim E \cdot 10^{-9}$) to avoid ill-conditioning of the stiffness matrix. The choice of the penalization factor (q) has a large influence on the optimization process. Choosing q too low will result in a gray solution, i.e. a solution with a lot of intermediate densities. While a q that is too high, will result in premature convergence to a local minimum. The 'best' value for the penalization factor resulting in near 0 – 1 solutions is 3 [64].

The FCM allows for material interpolation using the SIMP method. The stiffness penalization of the fictitious domain using α is replaced by the SIMP interpolation, by slightly rewriting Equation 2.33 into:

$$\mathbf{k}_c = \sum_{i=1}^{n_{sc}} E_{min} + \rho_i^q (E - E_{min}) \mathbf{k}_i \tag{2.49}$$

2.2.2 Gradient based optimization

The discretized optimization problem can be solved using several gradient-based optimization methods. Nowadays, the method of moving asymptotes (MMA) has become the standard for density based topology optimization problems [27, 69]. In this section the use of the MMA will be discussed, but, first a general method to obtain the gradient of the objective and constraint functions will be shown.

Sensitivity analysis using the adjoint analytical method

The adjoint analytical method is a general method to obtain the sensitivities of an objective or constraint function in an efficient manner [23]. Lets consider objective function \hat{F} , which is dependent on the design vector $\boldsymbol{\rho}$. The objective is also dependent on the coefficient vector \mathbf{D} , which is dependent on the density distribution:

$$\hat{F}(\boldsymbol{\rho}) = F(\boldsymbol{\rho}, \mathbf{D}) \quad (2.50)$$

To obtain an analytical expression of the sensitivity the chain rule has to be applied:

$$\frac{\partial \hat{F}}{\partial \rho_e} = \frac{\partial F}{\partial \rho_e} + \frac{\partial F}{\partial \mathbf{D}} \frac{\partial \mathbf{D}}{\partial \rho_e} \quad (2.51)$$

The derivative of the displacement field with respect to the density can be obtained by differentiating the finite element equation:

$$\frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{D} + \mathbf{K} \frac{\partial \mathbf{D}}{\partial \rho_e} = \frac{\partial \mathbf{F}}{\partial \rho_e} \quad (2.52)$$

which can be rewritten as:

$$\frac{\partial \mathbf{D}}{\partial \rho_e} = \mathbf{K}^{-1} \left(\frac{\partial \mathbf{F}}{\partial \rho_e} - \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{D} \right) \quad (2.53)$$

Substituting Equation 2.53 into Equation 2.51 yields:

$$\frac{\partial \hat{F}}{\partial \rho_e} = \frac{\partial F}{\partial \rho_e} + \frac{\partial F}{\partial \mathbf{D}} \mathbf{K}^{-1} \left(\frac{\partial \mathbf{F}}{\partial \rho_e} - \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{D} \right) \quad (2.54)$$

Now lets introduce the adjoint vector $\boldsymbol{\nu}$:

$$\boldsymbol{\nu}^T = \frac{\partial F}{\partial \mathbf{D}} \mathbf{K}^{-1} \quad (2.55)$$

In the adjoint method, one starts by solving the adjoint problem,

$$\mathbf{K} \boldsymbol{\nu} = \left(\frac{\partial F}{\partial \mathbf{D}} \right)^T \quad (2.56)$$

afterwards, $\boldsymbol{\nu}$ can be substituted into Equation 2.54 to obtain the desired sensitivity:

$$\frac{\partial \hat{F}}{\partial \rho_e} = \frac{\partial F}{\partial \rho_e} + \boldsymbol{\nu}^T \left(\frac{\partial \mathbf{F}}{\partial \rho_e} - \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{D} \right) \quad (2.57)$$

In a similar fashion the sensitivity of constraint function \hat{G}_i can be obtained.

The method of moving asymptotes (MMA)

The MMA optimization algorithm will be used to find the optimum material distribution for a given displacement field. The optimization problem in Equation 2.47 is appended to make sure there exists a feasible solution, even when a bad starting guess is chosen [69]:

$$\begin{aligned} \min_{\boldsymbol{\rho}} : & \hat{F}(\boldsymbol{\rho}) + a_0 z + \sum_{i=1}^m (c_i y_i) \\ \text{s.t.} : & \hat{G}_i(\boldsymbol{\rho}) - a_i z - y_i \leq 0, \quad i = 1, \dots, m \\ & : y_i \geq 0, \quad i = 1, \dots, m \\ & : \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1} \quad \forall \mathbf{x} \in \Omega \end{aligned} \quad (2.58)$$

where y_i , and z are 'artificial' optimization variables. Variables y_i are zero when the constraints are satisfied, and positive when the original optimization problem has no feasible solution. The corresponding coefficients c_i have to be 'sufficiently large' to allow the optimization algorithm to find a feasible solution, i.e. $c_i = 1000$. Coefficients a_i are generally set to zero, with $a_0 = 1$, such that z becomes zero. In the case that min-max optimization problems are considered, coefficients a_i have a finite values [69].

In the MMA, a convex sub-problem is created for each design iteration k . This convex sub-problem is an approximation of Equation 2.58 and is bounded by lower and upper asymptotes which are updated for each iteration. These asymptotes still require a starting guess, therefore, the lower asymptotes are initially set to be 0.001 while the upper ones start as 1.

In the sub-problems the design vector will be bounded by $\rho_{min}^{(k)}$ and $\rho_{max}^{(k)}$ which are defined as:

$$\rho_{min}^{(k)} = \max(\mathbf{0}, \rho^{(k)} - \zeta) \quad \rho_{max}^{(k)} = \min(\mathbf{1}, \rho^{(k)} + \zeta) \quad (2.59)$$

where ζ is a move-limit of 0.2. This move-limit will prevent oscillations in the design vector, thus making the optimization more stable.

Finally it has to be mentioned that both the objective and constraint functions have to be normalized. The objective function will be normalized with the value of the first guess, while the constraint is normalized with its constraint value.

2.2.3 Controlling the quality of the solution

It is well known that the continuous topology optimization problem defined in Equation A.1 lacks a solution [65]. In general, adding more holes to a structure will make it more efficient, hence reducing the objective function. In a similar fashion a fine discretized domain will include more holes than a coarse discretized domain. This problem is referred to as mesh-dependence [64].

Another well known problem in topology optimization are the infamous checkerboard patterns. The stiffness of these are alternating solid-void patterns, shown in Figure 2.4, is over-estimated by the finite element model when linear shape functions are used [28]. These artificially stiff structures are therefore preferred by the optimizer over more realistic structures. When higher-order shape functions are used the stiffness can be calculated more accurately and the checkerboard patterns are not observed.

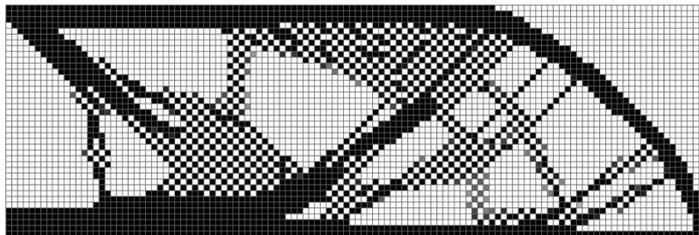


Figure 2.4: Artificially stiff checkerboard patterns on a MBB-beam optimization example

To cope with both mesh-dependence and checkerboard patterns different regularization methods exist. These methods have to somehow restrict the local or global variation of density. This can either be done by perimeter control, a gradient constraint or mesh-independent

filtering [65]. Mesh-independent filtering is the most efficient solution, as well as the most easy to implement, and most used method.

In this thesis five well-known mesh-independent filtering schemes are considered. These are: the sensitivity filter, the density filter, the density filter using Heaviside projection, the density filter using a modified Heaviside projection, and robust topology optimization. For the theory and implementation of morphology-based filters the reader is referred to [61].

Sensitivity filtering

The sensitivity filter introduced by Sigmund [59] modifies the voxel sensitivity based on the sensitivities of the surrounding voxels within a mesh-independent radius r_{min} . These filtered sensitivities are then used to update the design vector:

$$\widetilde{\frac{\partial \hat{F}}{\partial \rho_e}} = \frac{1}{\max(\rho_e, 0.001) \sum_{i=1}^{n_e} H_{ei}} \sum_{i=1}^{n_e} H_{ei} \rho_i \frac{\partial \hat{F}}{\partial \rho_i} \quad (2.60)$$

where n_e is the number of voxel, and H_{ei} is a linear decaying weighting function. The small number is put in the denominator to avoid division by zero. H_{ei} depends on the distance between the voxel center, and the center of the surrounding voxel, as well as the filter radius r_{min} :

$$H_{ei} = r_{min} - \text{dist}(e, i) \quad (2.61)$$

Density filtering

An alternative to the sensitivity filter is the density filter [19, 22]. The physical density of a voxel $\tilde{\rho}_e$ is defined as the weighted average of the design variables of neighboring voxels in r_{min} :

$$\tilde{\rho}_e = \frac{1}{\sum_{i=1}^{n_e} H_{ei}} \sum_{i=1}^{n_e} H_{ei} \rho_i \quad (2.62)$$

Using this filter method, $\tilde{\rho}$ is referred to as the physical densities, used to interpolate the voxel stiffness. Whilst ρ is referred to as the design vector. Using the chain rule the sensitivities with respect to the design variables can be obtained:

$$\widetilde{\frac{\partial \hat{F}}{\partial \rho_e}} = \sum_{i=1}^{n_e} \frac{\partial \hat{F}}{\partial \tilde{\rho}_i} \frac{\partial \tilde{\rho}_i}{\partial \rho_e} \quad (2.63)$$

which can be written as:

$$\widetilde{\frac{\partial \hat{F}}{\partial \rho_e}} = \sum_{i=1}^{n_e} \frac{\partial \hat{F}}{\partial \tilde{\rho}_i} \frac{H_{ei}}{\sum_{j=1}^{n_e} H_{ji}} \quad (2.64)$$

Density filtering with Heaviside projection

A disadvantage of both the sensitivity and density filter is that they introduce regions with intermediate densities. To cope with this a projection scheme base on a smoothed Heaviside function has been introduced by Guest et al [35]. In this approach the filtered densities ($\tilde{\rho}$) are

now referred to as the 'intermediate design vector'. The physical density $\bar{\rho}_e$ can be calculated using:

$$\bar{\rho}_e = 1 - e^{-\beta\tilde{\rho}_e} + \tilde{\rho}_e e^{-\beta} \quad (2.65)$$

where β controls the smoothness of the Heaviside function. For $\beta = 0$ the filter gives exactly the same output as the density filter, when β goes towards infinity the Heaviside function is approximated. The sensitivity of the objective function w.r.t. a design variable ρ_e can be written as:

$$\frac{\widetilde{\partial\hat{F}}}{\partial\rho_e} = \sum_{i=1}^{n_e} \frac{\partial\hat{F}}{\partial\bar{\rho}_i} \frac{\partial\bar{\rho}_i}{\partial\tilde{\rho}_i} \frac{\partial\tilde{\rho}_i}{\partial\rho_e} \quad (2.66)$$

Density filtering with a modified Heaviside projection

It has to be noted that the Heaviside projection provides a length-scale only on the solid material, but no length-scale on the void. This means that the physical density is very sensitive to a change in the design vector. To cope with this problem, Sigmund introduced the modified Heaviside filter, in which a length-scale is applied on the void [61]. The physical density can then be calculated as:

$$\bar{\rho}_e = e^{-\beta(1-\tilde{\rho}_e)} - (1 - \tilde{\rho}_e)e^{-\beta} \quad (2.67)$$

Robust topology optimization

Robust topology optimization is somewhat different from the filter methods described above. It has been introduced by Sigmund as a method to perform manufacturing tolerant topology optimization [62]. Small changes in manufacturing should not lead to large changes in functionality. Another positive effect of this method is that it is able to put a length-scale on both the solid and the void material, eliminating the longstanding problem of one-node connected hinges [78].

Instead of using the smoothed Heaviside projection either on solid or void, a threshold projection is used:

$$\bar{\rho}_e = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho}_e - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))} \quad (2.68)$$

where η is the threshold parameter. All filtered densities $\bar{\rho}_e$ above η are projected to 1, and all densities below to 0. It is important to note that for $\eta = 0$, the projection corresponds to the Heaviside projection, while $\eta = 1$ corresponds to the modified Heaviside projection.

In robust topology optimization three different designs are formulated based on the same design vector. These are a dilated ($\bar{\rho}^d$), intermediate ($\bar{\rho}^i$), and eroded ($\bar{\rho}^e$) design, with thresholds η , 0.5, and $1 - \eta$, respectively. The optimization problem is now reformulated as a min-max problem. The sensitivities can be obtained using Equation 2.66. Analogous to [78] the volume constraint is imposed on the dilated design, and updated every 20 iterations.

The three optimized designs will get similar topologies, such that the intermediate design is robust towards manufacturing errors. It is very important to note that in robust topology optimization an actual length-scale will be enforced on the final design ($\bar{\rho}^i$). The size of the length-scale depends on the choice of η . When $\eta = 0.25$, the length-scale exactly corresponds to the filter diameter [78].

2.3 FCMTOP: the Finite Cell Method Topology Optimization Program

The FCM and topology optimization methods described above are combined in FCMTOP. This program is based on FCMLab: A Finite Cell Research Toolbox for MATLAB, developed by Zander et al [85]. FCMLab is programmed in an object-oriented manner, making it easy to extend while maintaining the structure of the program. FCMTOP continues on this object-oriented set-up, allowing it to model very diverse topology optimization problems, specified by well-defined parameters.

This section will start by a description of the capabilities of FCMTOP to model diverse optimization examples, followed by a description of the two most important optimization examples. The section will be concluded by a schematic overview of the how topology optimization is performed by FCMTOP.

2.3.1 Capabilities of FCMTOP

The capabilities of FCMTOP to define an optimization example are shown in Table 2.1. For minimum compliance problems multiple load-cases can be present, when this is the case, either a Pareto optimum is found, or min-max optimization is performed. All other important parameters, such as the discretization $(n_{e,x}, n_{e,y}, n_{e,z})$, mesh dimensions (L_x, L_y, L_z) , penalization factor (q) , volume fraction (V_{max}) , polynomial degree (p) , and the number of voxels per cell in each direction (n_{voxel}) have to be defined by the user.

Table 2.1: Overview of the capabilities of FCMTOP

Three different objectives	Minimum compliance Minimum displacement Maximum displacement
Two types of constraints	Volume constraint (mandatory) Displacement constraints
Five different filters	Sensitivity filter Density filter Density filter with Heaviside projection Density filter with modified Heaviside projection Robust topology optimization
Three constitutive relations	Plane stress Plane strain 3D
Dirichlet BCs	Strong sense, using penalty method Weak sense, using Nitsche's method
Neumann BCs	Strong sense, applied on cell boundary Weak sense, using boundary parameterization
Nodal springs	Added to cell nodes
Passive elements	Both solid and void

2.3.2 Optimization problems

In the research in this thesis two optimization problems will be used extensively. These are the MBB-beam, and the compliant force inverter. In this section the parameterization of these problems will be discussed.

MBB-beam

The optimization problem for the MBB-beam example is written as:

$$\begin{aligned}
 \min_{\boldsymbol{\rho}} : \hat{F}(\boldsymbol{\rho}) &= \mathbf{D}^t \mathbf{K} \mathbf{D} \\
 \text{s.t.} : \hat{G}_1(\boldsymbol{\rho}) &= \frac{\sum \boldsymbol{\rho}}{n_e V_{max}} - 1 \leq 0 \\
 : \mathbf{0} \leq \boldsymbol{\rho} \leq 1 & \quad \forall \mathbf{x} \in \Omega
 \end{aligned} \tag{2.69}$$

The corresponding design domain and its dimensions are shown in Figure 2.5, due to symmetry only half of the design domain is used. The material properties of steel are used, $E = 210$ GPa, and $\nu = 0.3$, and the load has a magnitude of 1 MN. Plane stress conditions are assumed where domain thickness $t = 1$ mm. For the optimization a penalization factor $q = 3$, while the maximum allowable volume fraction $V_{max} = 0.4$. The discretization $(n_{e,x}, n_{e,y})$, the polynomial degree p , the number of voxels per cell in each direction (n_{voxel}), and the filter method are now the only parameters that can be varied in the upcoming experiments.

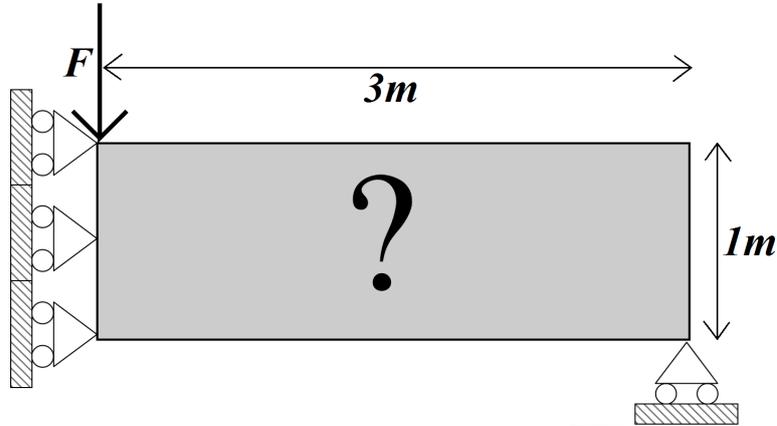


Figure 2.5: Design domain and boundary conditions for the MBB-beam problem

Using the adjoint analytical method the unfiltered sensitivity of the objective can easily be determined:

$$\boldsymbol{\nu}^T = \mathbf{F}^T \mathbf{K}^{-1} = \mathbf{D}^T \tag{2.70}$$

$$\frac{\partial \hat{F}}{\partial \rho_e} = -\mathbf{d}_c^T \left(\frac{\partial \mathbf{k}_c}{\partial \rho_e} \mathbf{d}_c \right) \tag{2.71}$$

where,

$$\frac{\partial \mathbf{k}_c}{\partial \rho_e} = q(E - E_{min})\rho_e^{q-1} \mathbf{k}_e^0 \tag{2.72}$$

The unfiltered sensitivity of the volume constraint is defined as:

$$\frac{\partial \hat{G}_1}{\partial \rho_e} = \frac{1}{n_e V_{max}} \quad (2.73)$$

Finally, it has to be noted that both the objective and its corresponding sensitivities are normalized by the compliance of the first iteration.

Compliant force inverter

The compliant force inverter introduced by Sigmund [59] is a compliant mechanism. Here the goal is to transfer work from an input actuator to an output spring. Due to its strong local minimum at $u_{out} = 0$ the force inverter is a very interesting optimization example. The optimization problem for the force inverter is written as:

$$\begin{aligned} \min_{\boldsymbol{\rho}} : \hat{F}(\boldsymbol{\rho}) &= \mathbf{L}^T \mathbf{D} \\ \text{s.t.} : \hat{G}_1(\boldsymbol{\rho}) &= \frac{\sum \boldsymbol{\rho}}{n_e V_{max}} - 1 \leq 0 \\ &: \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1} \quad \forall \mathbf{x} \in \Omega \end{aligned} \quad (2.74)$$

where \mathbf{L} is a vector with zeros at all degrees of freedom except at the output point, where its value is one. The corresponding design domain and its dimensions are shown in Figure 2.6, due to symmetry only half of the design domain is used.

Finding the right settings for this optimization problem proved to be more difficult than expected. The reason is the weak Dirichlet boundary conditions in the bottom left corner of the design domain. Its enforcement requires integration over a parameterized boundary using the same shape functions as the FE system, resulting in large differences between $p = 1$, and $p = 5$. To get a fair comparison between the experiments for different n_{voxel} , and p , the influence of different p on the boundary condition should be as small as possible. By trial and error, values for Γ_D , τ_N , E , and F , have been found that have the least influence on the optimized structures.

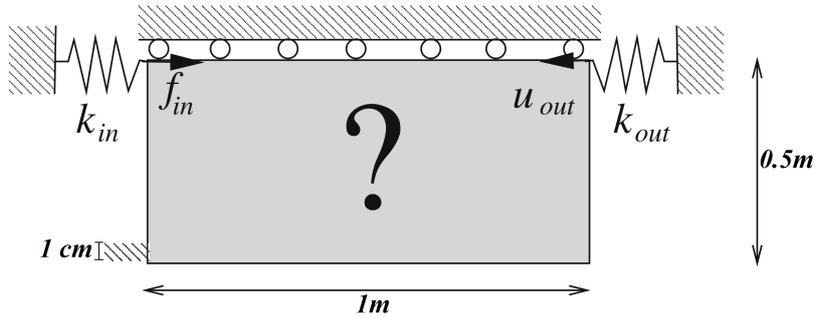


Figure 2.6: Design domain and boundary conditions for the compliant force inverter problem

The Young's modulus $E = 100$ Pa, $\nu = 0.3$, the load has a magnitude of 1 N, and the stability parameter $\tau_N = 500$. Plane stress conditions are assumed with a unit domain thickness. The input spring stiffness $k_{in} = 100$, and the output spring stiffness $k_{out} = 0.1$.

For the optimization a penalization factor $q = 3$, while the maximum allowable volume fraction $V_{max} = 0.3$. The discretization $(n_{e,x}, n_{e,y})$, the polynomial degree p , the number of voxels per cell in each direction (n_{voxel}) , and the filter method are now the only parameters that can be varied in the upcoming experiments.

Using the adjoint analytical method the unfiltered sensitivity of the objective can easily be determined:

$$\boldsymbol{\nu}^T = \mathbf{L}^T \mathbf{K}^{-1} \quad (2.75)$$

$$\frac{\partial \hat{F}}{\partial \rho_e} = -\boldsymbol{\nu}_c^T \left(\frac{\partial \mathbf{k}_c}{\partial \rho_e} \mathbf{d}_c \right) \quad (2.76)$$

Contrary to the MBB-beam both the objective and its corresponding sensitivities are not normalized by the output displacement of the first iteration. A computational trick is applied where the objective and sensitivities are normalized by the output displacement at each iteration. By doing so, the sub-problem in the MMA becomes highly sensitive to small changes when the output displacement is close to zero. Experiments performed have shown that this trick makes sure that the optimizer gets past the strong local minimum at $u_{out} = 0$.

Finally, it has to be noted that when robust topology optimization is applied, only the intermediate design $(\bar{\rho}^i)$ will be used for the objective. This is done in order to prevent instabilities when the hinge is created. After 50 iterations min-max optimization is applied to obtain a robust design.

2.3.3 The optimization process

A schematic overview of how topology optimization works with FCMTOP can be seen in Figure 2.7. The implementation of the optimization loop is based on the 88-line MATLAB code by Andreassen et al [10], such that the core parts of FCMTOP are the optimization functions. This is where the most of the time of the optimization is spend.

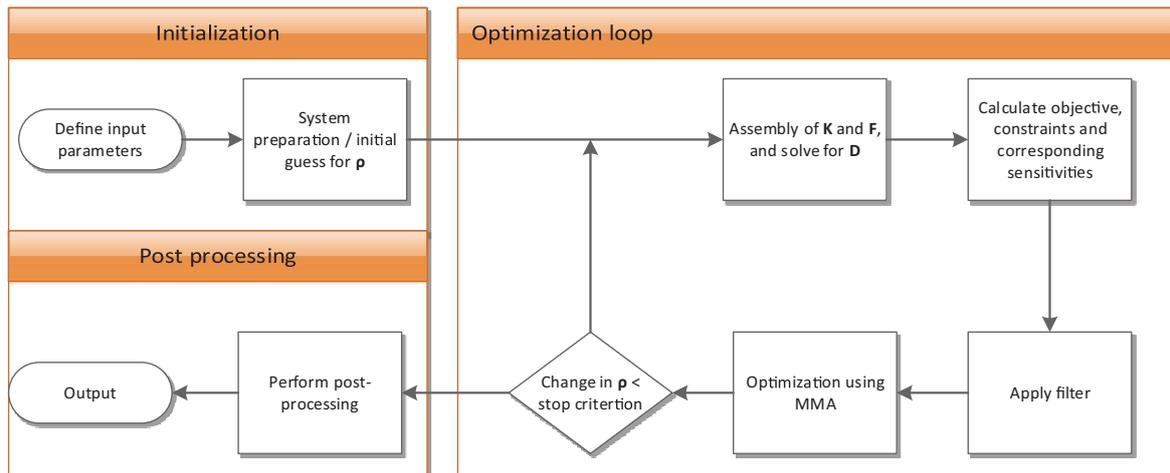


Figure 2.7: Schematic overview of the optimization process

An optimization starts by parameterization of the problem. FCMTOP then prepares the optimization, all functions and values that can be pre-computed before the actual optimization starts will be defined. The starting guess is defined such that all $\rho_i = V_{max}$. It is well known

that a different starting guess can have a large influence on the optimization, however, this is not considered in the research in this thesis.

The optimization is performed in a `while` loop until the stopping criterion is satisfied. Here the stopping criterion is defined as:

$$\max(\|\rho^{(k)} - \rho^{(k-1)}\|) \leq 0.01 \quad \text{or} \quad k > 1000 \quad (2.77)$$

Despite the fact that this stopping criterion does not say anything about the optimality of the objective, it is widely used [10, 60]. The reason for this is the stability as well as the ease of implementation.

When the stopping criterion is satisfied the post-processing is performed. In the first part of the post-processing the objective is re-calculated in a more accurate analysis. This is done to obtain a fair comparison between optimized structures at different settings, and to identify over-estimation of the stiffness as will be discussed in the next Chapter. In the second part of the post-processing process, the optimized structure, and its corresponding stress and displacement fields are visualized.

The validity of topology optimization using the FCM

In this chapter the validity of the Finite Cell Method Topology Optimization Program (FCM-TOP) regarding reliability and accuracy will be discussed. The focus of the research is put on the number of voxels per cell in each direction (n_{voxel}), the polynomial degree (p), and the filter method. Specific bounds on these variables have to be found ensuring general applicability of the optimization method.

First the limits of the FCM will be demonstrated, and, the need for a minimum length-scale will be shown. In the remainder of this chapter the numerical experiments with the MBB-beam and compliant force inverter will be discussed to find n_{voxel} , p and the filter methods that produce stable results.

3.1 The limits of FCMTOP

It is well known that a sufficient polynomial degree is required to model the variations within a cell. To demonstrate this, the MBB-beam example will be used with no filter, 5 by 5 voxels per cell ($n_{voxel} = 5$), a polynomial degree of 3, and a discretization of 360 by 120 voxels. In Figure 3.1, the corresponding density distribution, can be seen, where the gray lines correspond to the cell boundaries.

These settings produce even worse results than checkerboard-like patterns. The shape functions are not able to capture the variations within a cell, making the cells artificially stiff. The corresponding sensitivities are not correct resulting in an unrealistic 'optimized' design. To cope with this a higher polynomial degree is required, but, even when this is done it is very difficult to avoid these patterns, when no filter method is used.

To demonstrate this problem, consider the clamped beam shown in Figure 3.2, where a single layer of voxels has to represent a gap of void material. A uniform pressure load of 10 MN/m is applied at the right edge of the domain. Plane stress conditions are assumed, with a domain thickness of 1 cm. The material properties of steel are used with $E = 210$ GPa, the Young's modulus corresponding to the void is 210 Pa. A discretization of in total 200 by 100 voxels is used.

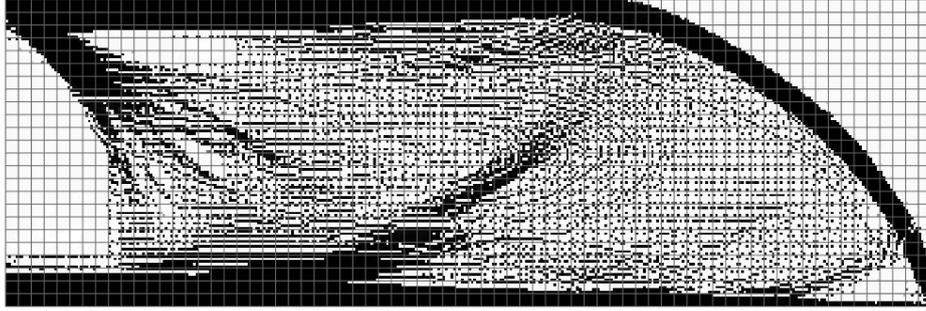


Figure 3.1: Artificially stiff 'optimized' MBB-Beam, no filter is used, $n_{voxel} = 5$, $p = 3$, and $V_{max} = 0.4$

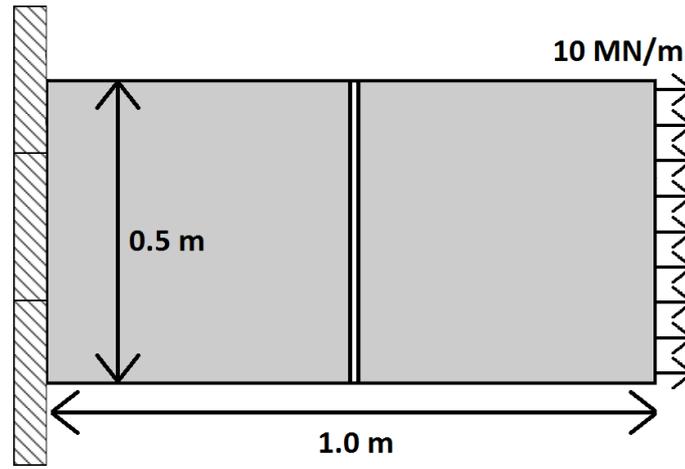


Figure 3.2: Split domain experiment to show the error in the displacement field, when a low polynomial degree is used

In Figure 3.3, the displacement in x -direction can be seen at the location of the gap. The first figure, is created using $n_{voxel} = 1$, and $p = 1$. Here, the model is able to identify a large discontinuity in the displacement. In the right figure with $n_{voxel} = 5$, and $p = 5$ the shape functions cannot describe this discontinuity, hence, the cell is artificially stiff.

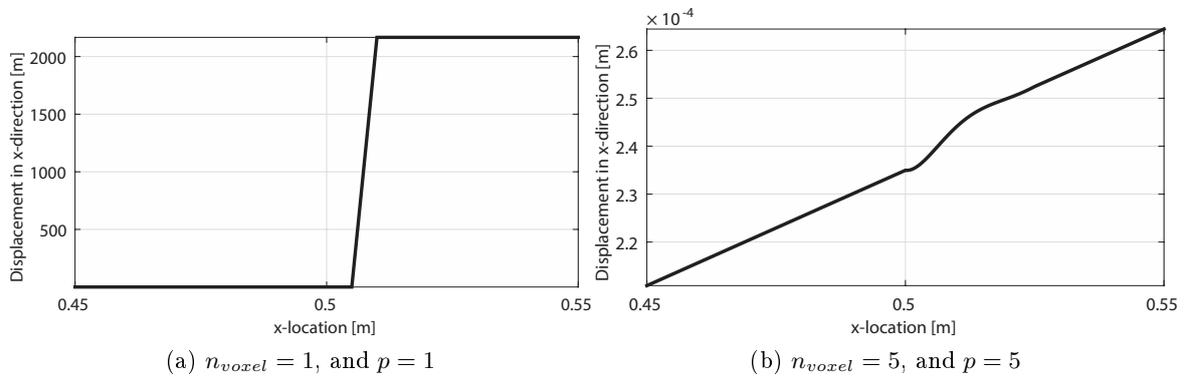


Figure 3.3: The displacement in x -direction at the location of the void

Experiments with $n_{voxel} > 1$, and different polynomial degrees have shown that the FCM is

not able to calculate the jump in displacement over a single layer of void material. For a wider strip of void material, the discontinuity in displacement can be identified at a reasonably low polynomial degree. Thus, a length-scale has to be applied on both the solid and void material. To indicate this effect, the jump in displacement is shown for a 3 voxel-wide strip of void material, in Figure 3.4, where $p = 3$, and $n_{voxel} = 5$. In the first case, the void is located at the first three columns of a cell. This allows the shape functions to describe a large jump in displacement. In the second case the void is located at the second, third and fourth column of the cell. Here, the jump in displacement is significantly smaller.

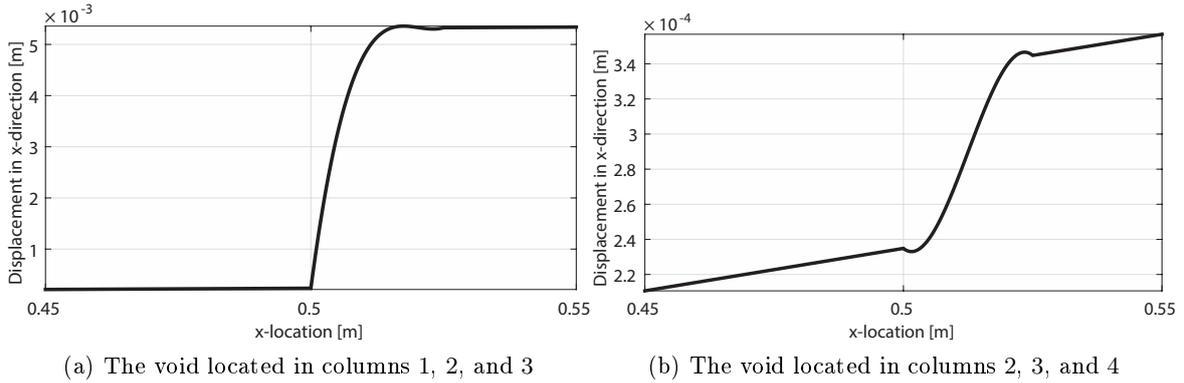


Figure 3.4: The displacement in x -direction at the location of the three voxel wide strip of void $n_{voxel} = 5$, and $p = 3$

Despite the fact that the discontinuity in displacement in the second figure is small, it is significant enough to affect the compliance. It is very important to realize that for topology optimization the analysis of an artificially stiff cell does not have to be exact, but, sufficient enough such that changes in cell stiffness are detected. This will make sure that the optimizer will go to more realistic cell topologies. It is well known that for smooth structures with realistic cell topologies, analysis with the FCM has a higher accuracy than classical FEM with linear shape functions [31, 52, 58].

A length scale imposed using mesh-independent filters can prevent these undesired cell topologies. To show this, the sensitivity filter is applied to the MBB-beam example, shown in Figure 3.1. If a filter radius (r_{min}) of two voxel lengths is chosen the obtained topology is exactly the same as when linear FEM is used. Both optimized structures can be seen in Figure 3.5. It is very important to note that the objective in both optimizations is calculated using a different analysis mesh. To allow a fair comparison all optimized objectives are recalculated using $n_{voxel} = 1$, and $p = 3$.

When the filter radius is increased, even lower polynomial degrees can be used. For linear shape functions and a sufficient filter radius the obtained results become similar to the multi-resolution topology optimization (MTOP) method, developed by Nguyen et al [47, 48]. It is very important to know that this decrease in computational complexity comes at a cost. A larger filter radius will introduce a length-scale spanning of at least one cell-width, hence, the gain in resolution is not exploited to the fullest.

In the remainder of this chapter, the parameters (n_{voxel} , and p) which produce stable results, are found for all implemented filters, with $r_{min} = 2$ voxel-widths.

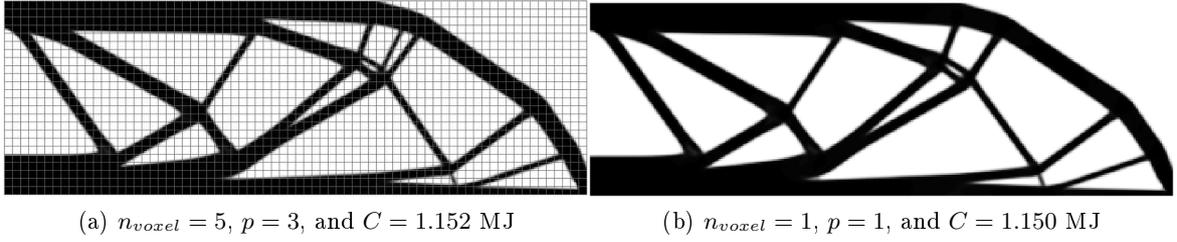


Figure 3.5: The MBB-Beam example for two different settings, both values of the compliance C are obtained with $n_{voxel} = 1$, and $P = 3$. A discretization of 360 by 120 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

3.2 Stable parameters for minimum compliance problems

The MBB-beam example will be used to obtain the relations between n_{voxel} and p for which the method produces stable results in minimum compliance optimization. To do so, different experiments have been performed for two different discretizations, (180 by 60), and (240 by 80). This allows experiments for different n_{voxel} (1, 2, 3, 4, 5, 6, 8 and 10), at polynomial degrees ranging from 1 to 5. All these experiments are performed for the five different filters, using $r_{min} = 2$ voxel widths, resulting in a total amount of 325 optimizations. The filter radius is not mesh-independent, but, the goal is not to get the exact same topologies, it is to find which settings FCMTOP produces realistic density distributions.

The results of all these experiments, which are the number of iterations (n_{iter}), objective ($\hat{F}(\boldsymbol{\rho})$), compliance (C), and measure of non-discreteness (M_{nd}) can be found in Appendix B. The measure of non-discreteness has been introduced by Sigmund to quantify the amount of gray material in the optimized design [61], and can be calculated using:

$$M_{nd} = \frac{\sum_{n_e} 4\bar{\rho}_e(1 - \bar{\rho}_e)}{n_e} \times 100\% \quad (3.1)$$

Due to the large number of combinations of parameters not all experiments have produced stable results. An optimized structure is regarded as valid when the following three conditions hold:

- No artificially stiff cells should be present, i.e. all cells should have a smooth topology
- The difference between the objective, and the post-processed compliance should be small
- The difference in density distribution between the optimized structure, and the known optimum should be small

All settings that produce any sort of artificially stiff cell will be regarded as not valid. This is also the case in the optimization example with $n_{voxel} = 5$, and $p = 2$, shown in Figure 3.7. At two locations the structure seems to be disconnected. Due to the gray material the structure is not completely disconnected, hence the difference between compliance and the objective is relatively small. Nevertheless, structures with visual flaws will be regarded as not valid.

The over-estimation of the cell stiffness, actually shows that the underlying optimization algorithm works well. Consider four different experiments ($n_{voxel} = 1$, $n_{voxel} = 3$, $n_{voxel} = 4$,

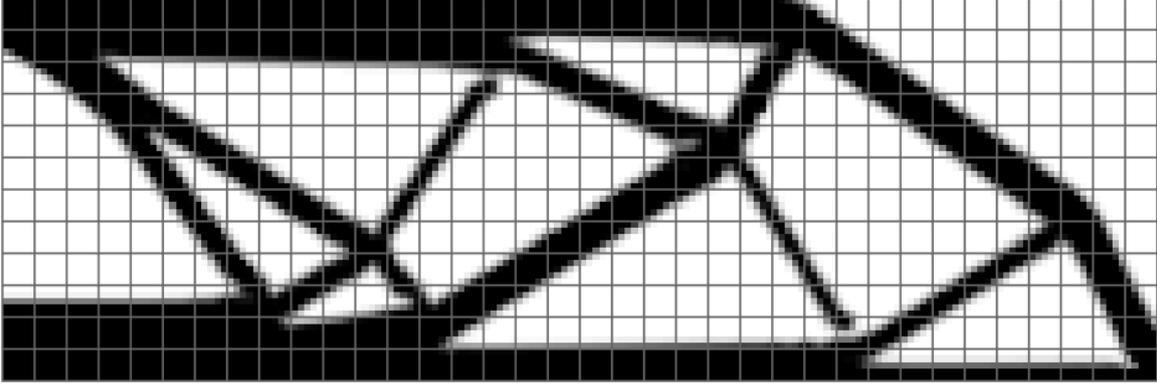


Figure 3.6: The MBB-Beam example, $n_{voxel} = 5$, $p = 2$, $\hat{F}(\rho) = 1.11\text{MJ}$, and $C = 1.20\text{MJ}$. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

and $n_{voxel} = 10$) shown in Figure 3.6, where the gray lines correspond to the cell boundaries. The optimized structure at $n_{voxel} = 3$ is not valid due to the artificially stiff cell, but, overall the optimized topology looks similar to the one of $n_{voxel} = 1$. If the optimizer is performing an optimization using the FCM with $n_{voxel} = 3$ and $p = 1$, the objective can become lower using these artificial stiff cells. If a smooth structure is modeled using the same analysis mesh ($n_{voxel} = 3$ and $p = 1$) the corresponding objective is slightly higher.

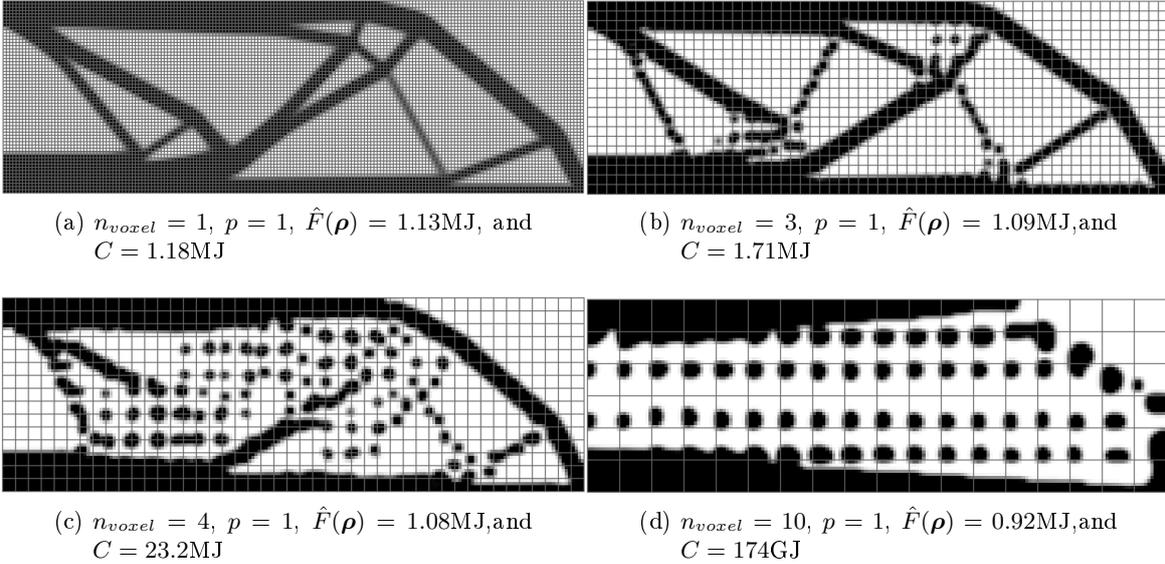


Figure 3.7: The MBB-Beam example for different settings. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

It is interesting to see that for $n_{voxel} = 10$, and $p = 1$, the cell nodes which affect the stiffness the most contain material. Due to the linear shape functions only 4 nodes affect the stiffness therefore placing material in the middle of a cell is not beneficial.

The different experiments with the sensitivity filter are summarized in Table 3.1. The green cells correspond to parameters that produce smooth structures, while the red cells correspond to invalid structures. The compliances (C) are shown, and it can be seen that for the

settings producing valid structures, the corresponding compliance is almost constant. Some experiments are modeled at both discretizations, in this case, the compliance of the worst performing experiment is shown. Finally, it has to be noted that the experiments for $n_{\text{voxel}} = 1$ & $p > 2$ are not performed, the corresponding Finite Cell mesh is very large, hence, the optimization will take too much time.

Table 3.1: Results of the stability experiments using sensitivity filtering, the compliance (C) is shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	1.18 MJ	1.19 MJ			
$n_{\text{voxel}} = 2$	1.18 MJ	1.19 MJ	1.18 MJ	1.19 MJ	1.19 MJ
$n_{\text{voxel}} = 3$	1.71 MJ	1.18 MJ	1.18 MJ	1.18 MJ	1.18 MJ
$n_{\text{voxel}} = 4$	23.2 MJ	1.18 MJ	1.18 MJ	1.18 MJ	1.19 MJ
$n_{\text{voxel}} = 5$	37.3 MJ	1.20 MJ	1.18 MJ	1.18 MJ	1.18 MJ
$n_{\text{voxel}} = 6$	165 MJ	2.03 MJ	1.19 MJ	1.18 MJ	1.19 MJ
$n_{\text{voxel}} = 8$	308 MJ	36.8 MJ	1.16 MJ	1.16 MJ	1.16 MJ
$n_{\text{voxel}} = 10$	174 GJ	38.6 MJ	1.76 MJ	1.23 MJ	1.19 MJ

The results in this table are not definitive, i.e. not all the settings indicated as stable (green) will be stable in any case. Topology optimization is still a very complex process, depending on numerous parameters, therefore, the settings at which it is performed are highly problem dependent. It is interesting to note that the combination $p = 3$, and $n_{\text{voxel}} = 8$ produces a smooth structure while the corresponding stiffness matrix is significantly smaller than $p = 1$ and $n_{\text{voxel}} = 1$. There is a high probability that these settings ($p = 3$, and $n_{\text{voxel}} = 8$) will fail to produce stable results in an optimization example with more complex boundary conditions.

Similar tables have been created for the other four types of filters, they can be found in Tables 3.2- 3.5.

Table 3.2: Results of the stability experiments using density filtering, the compliance (C) is shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	1.22 MJ	1.21 MJ			
$n_{\text{voxel}} = 2$	1.22 MJ	1.22 MJ	1.21 MJ	1.21 MJ	1.21 MJ
$n_{\text{voxel}} = 3$	1.58 MJ	1.22 MJ	1.22 MJ	1.21 MJ	1.21 MJ
$n_{\text{voxel}} = 4$	46.8 MJ	1.22 MJ	1.22 MJ	1.22 MJ	1.21 MJ
$n_{\text{voxel}} = 5$	54.8 MJ	1.21 MJ	1.22 MJ	1.22 MJ	1.21 MJ
$n_{\text{voxel}} = 6$	64.9 MJ	3.68 MJ	1.21 MJ	1.22 MJ	1.21 MJ
$n_{\text{voxel}} = 8$	74.3 MJ	64.1 MJ	1.23 MJ	1.19 MJ	1.19 MJ
$n_{\text{voxel}} = 10$	141 GJ	52.7 MJ	37.5 MJ	1.26 MJ	1.21 MJ

As has been said it is very important to remind that the distinction between stable (green), and unstable (red) settings, has to be interpreted as a recommendation on which settings to use. Nevertheless, some important conclusions can be drawn based on the experiments with the MBB-beam example. The first is that for each filter method, the optimized structures that are regarded as stable, look visually almost identical. This might be due to the simplicity of the optimization example, however, it shows the general applicability of the method when different settings are used.

Table 3.3: Results of the stability experiments using density filtering with Heaviside projection, the compliance (C) is shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	1.18 MJ	1.12 MJ			
$n_{\text{voxel}} = 2$	1.16 MJ	1.19 MJ	1.12 MJ	1.17 MJ	1.15 MJ
$n_{\text{voxel}} = 3$	47.7 MJ	1.44 MJ	1.16 MJ	1.18 MJ	1.15 MJ
$n_{\text{voxel}} = 4$	43.2 MJ	1.20 MJ	1.14 MJ	1.18 MJ	1.16 MJ
$n_{\text{voxel}} = 5$	39.1 MJ	18.7 MJ	1.20 MJ	1.15 MJ	1.17 MJ
$n_{\text{voxel}} = 6$	53.9 MJ	44.5 MJ	1.48 MJ	1.16 MJ	1.17 MJ
$n_{\text{voxel}} = 8$	51.4 MJ	49.6 MJ	16.0 MJ	1.11 MJ	1.13 MJ
$n_{\text{voxel}} = 10$	27.7 GJ	41.0 MJ	49.3 MJ	3.32 MJ	1.44 MJ

Table 3.4: Results of the stability experiments using density filtering with modified Heaviside projection, the compliance (C) is shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	1.15 MJ	1.15 MJ			
$n_{\text{voxel}} = 2$	1.16 MJ	1.15 MJ	1.15 MJ	1.15 MJ	1.15 MJ
$n_{\text{voxel}} = 3$	3.05 MJ	1.16 MJ	1.15 MJ	1.14 MJ	1.14 MJ
$n_{\text{voxel}} = 4$	36.2 MJ	1.18 MJ	1.15 MJ	1.15 MJ	1.14 MJ
$n_{\text{voxel}} = 5$	29.1 MJ	1.48 MJ	1.16 MJ	1.15 MJ	1.15 MJ
$n_{\text{voxel}} = 6$	30.5 MJ	40.5 MJ	1.17 MJ	1.15 MJ	1.15 MJ
$n_{\text{voxel}} = 8$	27.0 MJ	30.9 MJ	1.61 MJ	1.20 MJ	1.15 MJ
$n_{\text{voxel}} = 10$	860 GJ	30.7 MJ	39.3 MJ	1.82 MJ	1.22 MJ

Table 3.5: Results of the stability experiments using robust topology optimization ($\eta = 0.2$), the compliance (C) is shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	1.21 MJ	1.20 MJ			
$n_{\text{voxel}} = 2$	1.23 MJ	1.21 MJ	1.21 MJ	1.21 MJ	1.22 MJ
$n_{\text{voxel}} = 3$	1.39 MJ	1.20 MJ	1.21 MJ	1.21 MJ	1.22 MJ
$n_{\text{voxel}} = 4$	1.76 MJ	1.21 MJ	1.20 MJ	1.20 MJ	1.20 MJ
$n_{\text{voxel}} = 5$	59.0 MJ	1.33 MJ	1.22 MJ	1.21 MJ	1.20 MJ
$n_{\text{voxel}} = 6$	55.2 MJ	17.1 MJ	1.19 MJ	1.21 MJ	1.20 MJ
$n_{\text{voxel}} = 8$	413 GJ	53.5 MJ	1.23 MJ	1.26 MJ	1.26 MJ
$n_{\text{voxel}} = 10$	3.53 TJ	60.9 MJ	1.57 MJ	1.41 MJ	1.20 MJ

As is expected the modified Heaviside projection method is more stable for a less complex analysis mesh than the Heaviside projection method. Density and sensitivity filtering both can perform stable topology optimization for even less complex analysis system. These two filtering methods, allow the use of material with intermediate density at locations in the structure where the need for solid elements is not required. An example can be seen in Figure 3.8, where the density filter is applied on the MBB-beam using $n_{\text{voxel}} = 4$, and $p = 2$. The majority of the thin structural members does not consist of solid elements. The penalized intermediate densities are more efficient than using solid elements, or even not using these structural members at all.

The projection filters suppress this gray material. This is done by making structural members

either completely solid or void, or in case a relatively simple FCA is used, the optimizer can exploit the artificially stiff cells. This effect can be seen in the right part of Figure 3.8, where a density filter using Heaviside projection is used.

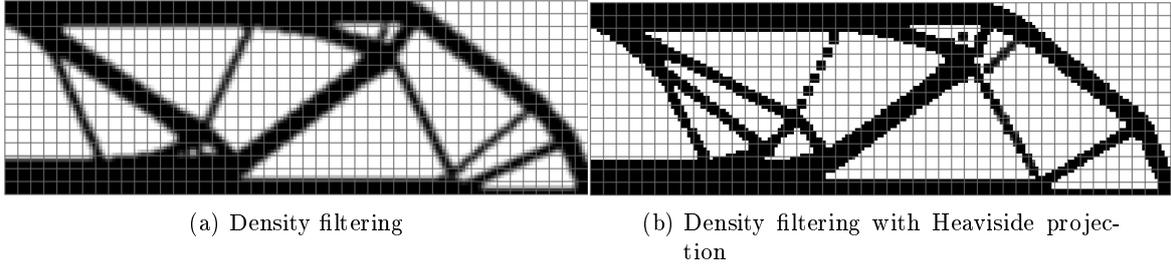


Figure 3.8: The MBB-Beam example for two different filter settings, with $n_{voxel} = 4$, and $P = 2$. A discretization of 180 by 60 voxels is used, and $r_{min} = 2$ voxel widths

Applying robust topology optimization creates crisp black and white structures, with a slightly higher compliance than when the projection methods are used. The reason for this is the imposed length-scale and the manufacturing tolerance. As is discussed by Sigmund, the self-adjoint nature of the minimum compliance problem makes sure that the eroded design always has the highest compliance, hence the effect of the robust formulation is not fully exploited [62]. The main advantage of using robust topology optimization in minimum compliance problems is the imposed length-scale on both solid and void compared to the projection methods.

3.3 Stable parameters for minimum displacement problems

In this section the compliant force inverter example will be used to obtain the relations between n_{voxel} and p for which the method produces stable results. This more challenging optimization example has been tested at two different discretizations, (120 by 60) and (160 by 80). The same range of p and n_{voxel} has been used as for the MBB-beam resulting in 325 different optimizations. The corresponding details of the optimizations can be found in Appendix C.

Compared to the MBB-beam example the results of the compliant force inverter example are more diverse, making it even more difficult to give conclusive remarks about the validity of the method for different settings. The discussion here should therefore be seen as a demonstration of what the capabilities of FCMTOP regarding minimum displacement problems, and not as a validation process. In this case the results of the experiments are regarded stable when the following two conditions hold:

- No artificially stiff cells should be present, i.e. all cells should have a smooth topology
- The difference between the objective, and the post-processed compliance should be small

In almost all performed experiments, hinges are created such that the objective becomes negative. The location of these hinges depends on the polynomial degree of the shape functions. When $p = 1$, or $p = 2$, the hinges are mostly located at cell nodes, such that the linear shape

functions will allow a hinge like deformation between the cells. Higher p will also allow the formation of hinges within a cell, which can be more efficient. For three different polynomial degrees, the optimized structures, and a closeup of the corresponding hinges can be seen in Figure 3.9.

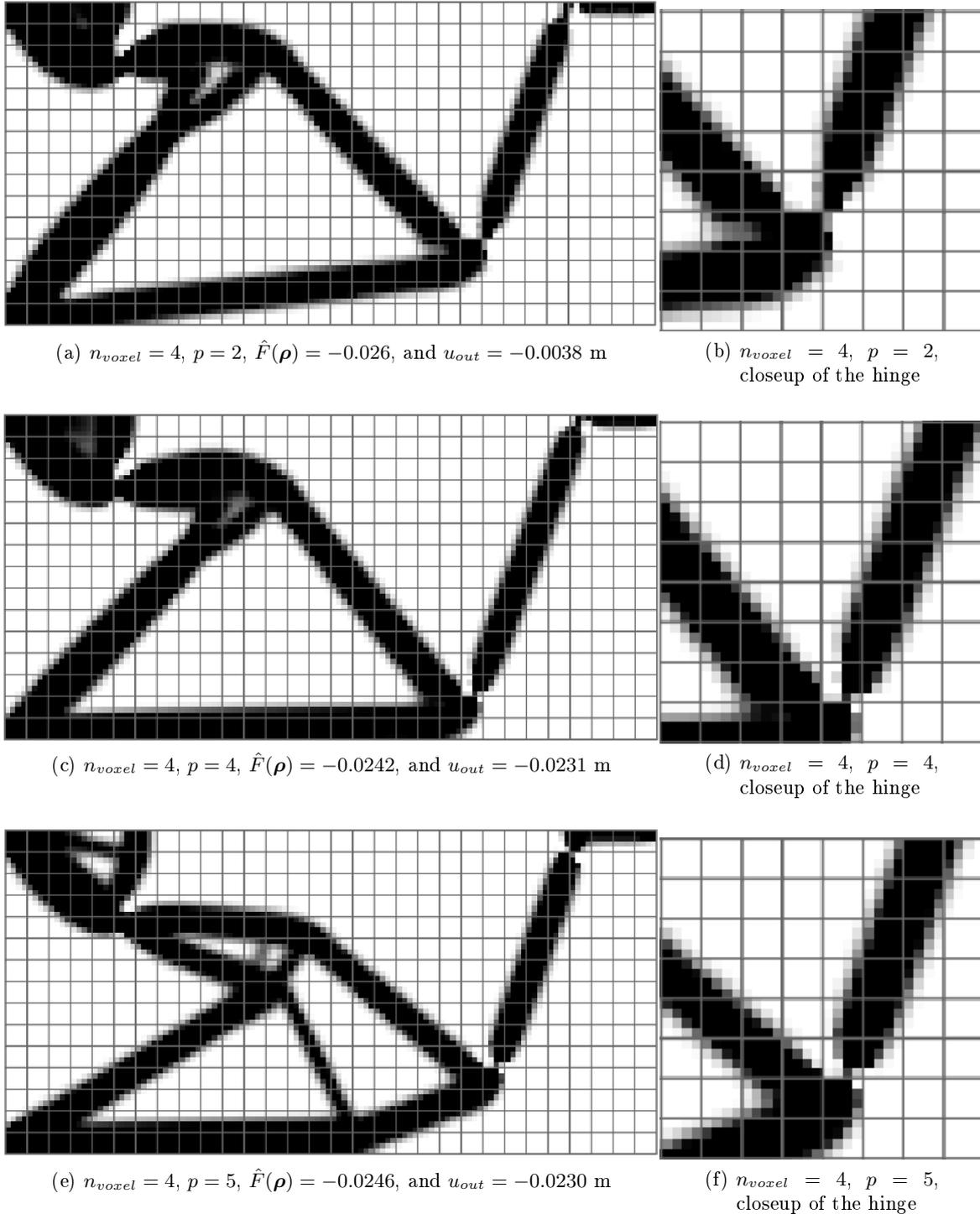


Figure 3.9: The compliant inverter example for three settings. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

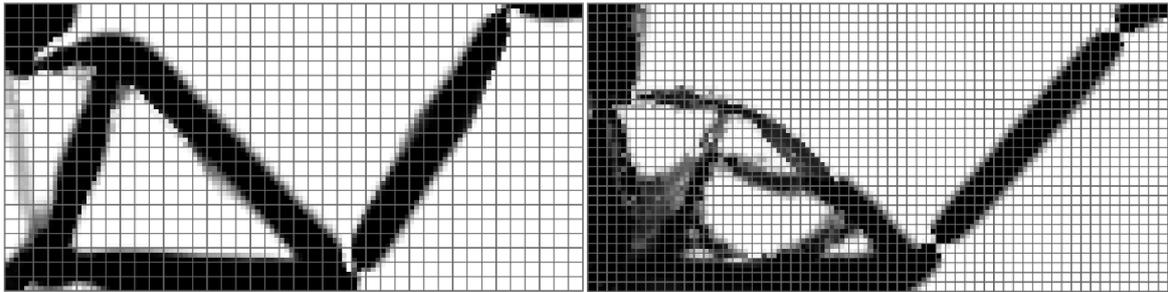
The optimized structure with $n_{voxel} = 4$ and $p = 2$, looks visually valid, however, it is not regarded as stable. A small disconnection in the upper right corner leads to a low output displacement, when recalculated using $n_{voxel} = 1$ and $p = 3$. It can also be seen that the structure optimized with $p = 4$, and the structure optimized with $p = 5$ look visually different, while their corresponding output displacements u_{out} are almost the same.

Similar to the MBB-beam example, tables have been created which summarize the minimum displacement experiments. These tables for all five implemented filter methods can be found in Tables 3.6- 3.10. In the case that an experiment is performed by 2 different discretizations, the results of the experiment that describes the behavior of the method best, will be shown.

Table 3.6: Results of the stability experiments using sensitivity filtering, the values for the output displacement u_{out} are shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	-2.16 cm	-1.99 cm			
$n_{voxel} = 2$	-1.14 cm	-2.26 cm	-2.33 cm	-2.30 cm	-2.28 cm
$n_{voxel} = 3$	-1.04 cm	-2.27 cm	-2.33 cm	-1.32 cm	-2.35 cm
$n_{voxel} = 4$	-1.54 cm	-0.38 cm	-2.28 cm	-2.31 cm	-2.30 cm
$n_{voxel} = 5$	0.00 cm	-1.85 cm	-2.28 cm	-2.32 cm	-2.13 cm
$n_{voxel} = 6$	0.00 cm	-1.3 cm	-2.25 cm	-2.28 cm	-2.11 cm
$n_{voxel} = 8$	0.00 cm	0.00 cm	0.00 cm	-2.35 cm	-2.30 cm
$n_{voxel} = 10$	0.00 cm	-0.06 cm	-0.52 cm	0.00 cm	-2.43 cm

The experiment with $n_{voxel} = 3$, and $p = 4$ is indicated with a different color (yellow). Nothing is wrong with the optimized structure, a negative displacement is obtained, and no artificially stiff cells are present, however, the output displacement is much smaller compared to other experiments. The reason for this is an instability at the beginning of the optimization leading to a connection between the input spring, and the weak boundary. As a result the optimized structure still consists partly of this connection as can be seen in Figure 3.10.



(a) $n_{voxel} = 3$, $p = 4$, $\hat{F}(\rho) = -0.0175$, and $u_{out} = -0.0132$ m

(b) $n_{voxel} = 2$, $p = 1$, $\hat{F}(\rho) = -0.0099$, and $u_{out} = -0.0114$ m

Figure 3.10: The compliant inverter for two settings where an instability at the beginning of the optimization causes a connection between the input spring and the weak boundary. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

This effect is not unique to these settings and has occurred in other experiments as well. In the experiment with $p = 1$, and $n_{voxel} = 2$, the same instability occurred, but, here the optimizer failed to get rid of this connection. Finally, it has to be noted that the occurrence

of this instability seems random, and that a small change in parameters or discretization will omit this effect.

Table 3.7: Results of the stability experiments using density filtering, the values for the output displacement u_{out} are shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	-2.19 cm	-1.99 cm			
$n_{voxel} = 2$	-2.14 cm	-2.21 cm	-1.87 cm	-2.20 cm	-2.19 cm
$n_{voxel} = 3$	-0.076 cm	-2.18 cm	-2.18 cm	-1.74 cm	-2.19 cm
$n_{voxel} = 4$	-1.76 cm	-2.11 cm	-2.16 cm	-2.19 cm	-2.18 cm
$n_{voxel} = 5$	0.00 cm	-2.15 cm	-2.22 cm	-2.21 cm	-2.14 cm
$n_{voxel} = 6$	0.00 cm	-1.96 cm	-2.18 cm	-2.18 cm	-2.19 cm
$n_{voxel} = 8$	0.00 cm	-2.15 cm	-2.29 cm	-2.15 cm	-2.28 cm
$n_{voxel} = 10$	0.00 cm	-1.52 cm	-1.31 cm	-2.15 cm	-2.20 cm

Table 3.8: Results of the stability experiments using density filtering with Heaviside projection, the values for the output displacement u_{out} are shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	-2.12 cm	-1.76 cm			
$n_{voxel} = 2$	-2.25 cm	-2.28 cm	-2.38 cm	-2.26 cm	-2.16 cm
$n_{voxel} = 3$	-1.52 cm	-2.26 cm	-2.28 cm	-2.19 cm	-2.17 cm
$n_{voxel} = 4$	0.00 cm	0.00 cm	-2.25 cm	-2.28 cm	-2.24 cm
$n_{voxel} = 5$	0.00 cm				
$n_{voxel} = 6$	0.00 cm	-1.27 cm	0.00 cm	-2.13 cm	0.00 cm
$n_{voxel} = 8$	0.00 cm	0.00 cm	0.00 cm	0.00 cm	-2.30 cm
$n_{voxel} = 10$	0.00 cm	0.00 cm	0.00 cm	0.00 cm	-1.46 cm

Table 3.9: Results of the stability experiments using density filtering with modified Heaviside projection, the values for the output displacement u_{out} are shown

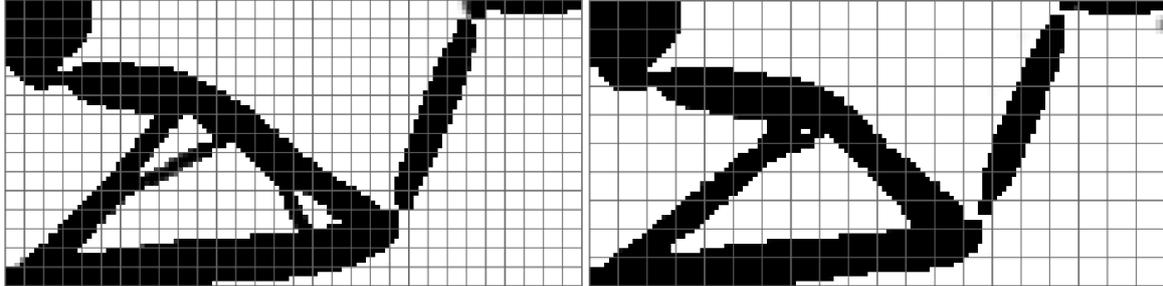
	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	-1.83 cm	-1.82 cm			
$n_{voxel} = 2$	-1.88 cm	-2.15 cm	-1.83 cm	-1.88 cm	-2.13 cm
$n_{voxel} = 3$	-1.96 cm	-2.02 cm	-1.98 cm	-1.53 cm	-2.09 cm
$n_{voxel} = 4$	0.00 cm	-1.21 cm	-2.13 cm	-2.00 cm	-2.09 cm
$n_{voxel} = 5$	0.00 cm	-0.30 cm	-2.13 cm	-1.98 cm	-1.77 cm
$n_{voxel} = 6$	-0.66 cm	-1.21 cm	-2.10 cm	-2.09 cm	-2.00 cm
$n_{voxel} = 8$	0.00 cm	0.00 cm	-2.02 cm	-2.12 cm	-1.74 cm
$n_{voxel} = 10$	0.00 cm	0.00 cm	-1.55 cm	-2.07 cm	-2.10 cm

Despite the fact that this optimization problem is more complex than the minimum compliance problem it can be seen that the FCMTOP produces stable results for similar ranges of settings. This emphasizes the general applicability of the method. It is interesting to see that in some cases settings where $p \neq 1$, and $n_{voxel} \neq 1$, the optimized structure has a higher u_{out} than when $p = 1$, and $n_{voxel} = 1$, are used. This makes these settings not better, however, it shows the complexity of the problem and the larger number of local minima at which the optimizer can get stuck. The main goal of this test example is to see whether the optimization method is able to create hinges, in which FCMTOP has succeeded.

Table 3.10: Results of the stability experiments using robust topology optimization ($\eta = 0.2$), the values for the output displacement u_{out} are shown

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	-1.95 cm	-1.60 cm			
$n_{voxel} = 2$	-1.81 cm	-1.97 cm	-1.35 cm	-1.79 cm	-1.88 cm
$n_{voxel} = 3$	0.00 cm	-2.03 cm	-1.62 cm	-1.70 cm	-1.90 cm
$n_{voxel} = 4$	0.00 cm	-2.00 cm	-1.88 cm	-1.78 cm	-1.81 cm
$n_{voxel} = 5$	0.00 cm	-1.87 cm	-2.13 cm	-1.98 cm	-1.77 cm
$n_{voxel} = 6$	-0.66 cm	-1.79 cm	-1.93 cm	-1.39 cm	-1.78 cm
$n_{voxel} = 8$	0.00 cm	0.00 cm	-2.10 cm	-1.62 cm	-1.78 cm
$n_{voxel} = 10$	0.00 cm	-1.42 cm	-1.82 cm	-1.83 cm	-1.85 cm

Most of the invalid structures have some sort of disconnection. The density filter with Heaviside projection filter is particularly prone to this effect. The projection method puts a length-scale on the solid material, therefore, one-node connected hinges are characteristic for this method. In case artificially stiff cells can be exploited this results in disconnected hinges, of which two examples can be seen in Figure 3.11. Especially in the experiments performed with the finer discretization (160 by 80) these disconnected hinges occur.



(a) $n_{voxel} = 4$, $p = 4$, $\hat{F}(\rho) = -0.0240$, and $u_{out} = -0.0228$ m

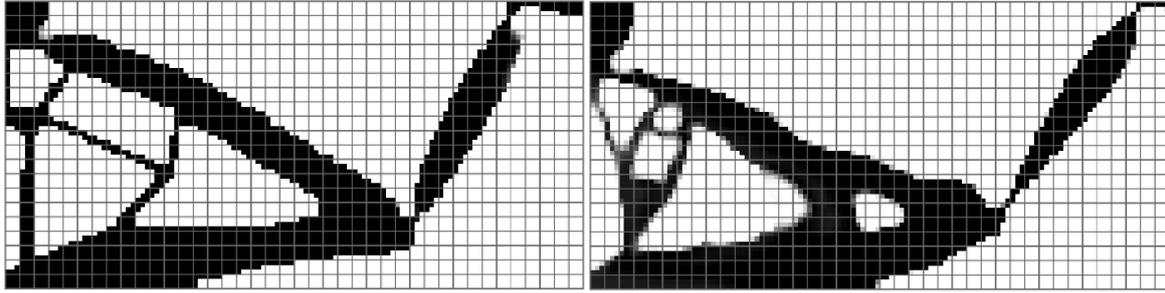
(b) $n_{voxel} = 6$, $p = 5$, $\hat{F}(\rho) = -0.0234$, and $u_{out} = 0.00$ m

Figure 3.11: The compliant inverter for two settings where the hinges are either connected by one node, or are completely disconnected. A discretization of 120 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths

It is interesting to note that in the left figure, the top right hinge consists of some gray material. In a few optimization examples with Heaviside projection hinges with intermediate density have appeared. The reason is again an artificially stiff cell, the density distribution, moves towards a disconnected hinge, but, a completely disconnected hinge has a negative effect on the objective. Therefore, the hinge will consist of intermediate densities.

In a similar fashion the modified Heaviside projection creates one-node connected hinges or disconnected hinges. So far the optimized results using this projection method seem to be performing worse than the optimized structures using the normal Heaviside projection method. Most optimized density distributions still show a connection between the weak boundary and the input spring, two examples of this can be seen in Figure 3.12. As has been discussed this is caused by an instability in the first iterations, after which it is hard to recover. The reason why this instability occurs more using the modified Heaviside projection

method is yet to be determined.



(a) $n_{voxel} = 3$, $p = 3$, $\hat{F}(\rho) = -0.0207$, and $u_{out} = -0.0198$ m

(b) $n_{voxel} = 3$, $p = 4$, $\hat{F}(\rho) = -0.0150$, and $u_{out} = -0.0153$ m

Figure 3.12: The compliant inverter for two settings where the connection between the weak boundary and input spring negatively influenced the objective. A discretization of 120 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths

Robust topology optimization is well suited for minimum displacement examples. The length-scale it imposes locally on both solid, and void material ensures no one-node connected hinges will be created. These thicker hinges result in a lower but more realistic output displacement. The manufacturing tolerant black and white structures, are created in a more stable manner than when a single projection method is used. The method seems to perform well for less complex analysis system compared to projection methods. This alleviates a part of the increased computational cost of having to solve the analysis equations three times.

A problem with robust topology optimization is that it seems to perform worse when finer discretizations are used, in combination with the small filter radius ($r_{min} = 2$). A fine discretization will result in more structural members, and the small filter radius will make sure that these members are extremely thin in the eroded design, hence making the optimization unstable.

3.4 Concluding remarks on the accuracy, and the stability of FCMTOP

From the experiments presented in this chapter it can be concluded that the FCMTOP is a stable and general applicable method. All experiments have been performed under the same conditions, to get a fair comparison. Better results can be achieved by tweaking the optimization parameters, e.g. the penalization factor, and move limit, per problem, however, that was not the goal of this study.

The optimized values for the compliance and the output displacement, differ per filter, p and n_{voxel} , these differences are small and no clear conclusion can be drawn on which setting is the best. In minimum compliance optimization the optimized structures regarded as stable look visually similar. In minimum displacement problems, the density distributions all end up at a slightly different local minima, however, the fact that hinges are created shows the capabilities of FCMTOP to handle complex optimization problems.

Finally, a very important aspect of the stability of the method is the amount of iterations n_{iter} required to reach convergence. A high number of iterations is not regarded as stable

since the optimization stops after 1000 iterations. This number is generally not reached, especially not when projection filters are used. The experiments have shown that a relation between the number of iterations, p and n_{voxel} is not apparent. Similar to the objective the amount of iterations differs per filter, p and n_{voxel} , and can be tweaked per problem. Most values of n_{iter} are in the same range, e.g. 300 – 450. It is impossible to say which setting produces the lowest number of iterations. An overview of the number of iterations required for different p and n_{voxel} can be seen in Table 3.11. For the other experiments the required number of iterations can be seen in the corresponding appendices.

Table 3.11: The amount of iterations per optimization of the MBB-beam for different settings. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	408	412	-	-	-
$n_{voxel} = 2$	422	210	509	437	330
$n_{voxel} = 3$	287	392	325	383	410
$n_{voxel} = 4$	289	649	378	365	285
$n_{voxel} = 5$	352	286	441	434	439
$n_{voxel} = 6$	194	339	349	388	273
$n_{voxel} = 10$	136	188	489	241	366

Efficient topology optimization using the FCM

In this chapter the computational cost of performing topology optimization using the Finite Cell Method (FCM) is discussed. More importantly, this cost is compared to the computational cost of performing topology optimization using classical FEM. Based on this comparison, the values for p and n_{voxel} are identified for which topology optimization using the FCM can result in an increase of the efficiency for topology optimization applications.

4.1 Drivers of computational cost in topology optimization

A topology optimization process consists of 3 steps: initialization, optimization, and post-processing. In this study on computational efficiency, only the optimization module will be considered. Although the computational cost of the initialization is highly dependent on p , and n_{voxel} , it should be only a fraction of the total computational cost. The time spent in the post-processor is independent of p , and n_{voxel} , and therefore will be left out of this discussion. The amount of iterations, n_{iter} , spent in the optimization module has a large influence on the computational cost of an optimization example. As has been discussed in the previous chapter, no relation could be found between n_{iter} , p , and n_{voxel} . The research on computational cost is therefore shifted to the time spent per design iteration.

4.1.1 The computational cost of one design iteration

A design iteration consists of:

1. Assembly of the stiffness matrix
2. Solving the analysis equations
3. Sensitivity analysis
4. Mesh-independent filtering

5. Design update using the MMA

Experiments performed with FCMTOP are in agreement with the theory that solving the analysis equations requires most computational effort [7]. The assembly of the stiffness matrix \mathbf{K} comes second regarding computational effort. The design update using the MMA uses a significant part of the computational cost as well, but, it is independent of p and n_{voxel} , therefore the corresponding computational effort will not be considered.

Assembly of the stiffness matrix

The assembly of the stiffness matrix includes: the mapping of the penalized stiffness on the pre-integrated cells, storage of the stiffness matrix, and the application of the boundary conditions. The most time consuming part of this process is storing the stiffness matrix using the `sparse` command in MATLAB. A relation exists between the memory required to store \mathbf{K} , and the corresponding time $T_{assembly}$. The required memory to store \mathbf{K} depends on both the number of non-zero elements n_{nz} and the size of the stiffness matrix, n_{dof} ,

$$bytes = (8 + 8) \times n_{nz} + 8 \times (n_{dof} + 1) \quad (4.1)$$

Solving the analysis equations

The unknown vector \mathbf{D} , is obtained using Cholesky decomposition. If a full matrix decomposed the computational cost is $O(n_{dof}^3)$, however, this is not the case since \mathbf{K} is sparse. The corresponding computational cost is therefore difficult to predict. It is known that this cost depends on both the half-bandwidth of the stiffness matrix (n_{bw}) and the degrees of freedom (n_{dof}) [26]. To keep the computational cost down, both the size of the stiffness matrix, and the bandwidth have to be kept small.

4.1.2 The size and shape of the stiffness matrix

Rudimentary relations have been identified for the computational cost of both the assembly of \mathbf{K} , and the corresponding Cholesky decomposition. It should be noted that these relations are simple and not completely accurate, however, they can give a decent overview on how the size and shape of \mathbf{K} influence the computational cost. Three parameters have been identified that drive this cost, that are the size of the stiffness matrix (n_{dof}), the half-bandwidth (n_{bw}), and the amount of non-zero elements (n_{nz}).

The size of the stiffness matrix

The size of \mathbf{K} , depends on p , and the number of cells n_c . For a cell with a given set of basis functions, the amount of corresponding unknowns can be determined. In a similar fashion the size of the unknown vector \mathbf{D} , which corresponds to n_{dof} , can be calculated. This size, as a function of p , and n_{voxel} can be seen in Table 4.1. In this example a discretization of 120 by 120 voxels is used, however, the relative differences hold for any type of discretization.

It is very interesting to see that n_{dof} is constant for all $p = n_{voxel}$, as well as for all $p = n_{voxel}/i$, where i can be any integer. This means that a higher polynomial degree does not necessarily have to leave to more degrees of freedom in the analysis system. Furthermore, these exact same relations hold when a 3D optimization example is considered.

Table 4.1: The size of the stiffness matrix (n_{dof}), for different polynomial degrees (p), and different cell topologies. A discretization of 120 by 120 voxels is used

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	29,282	116,162	260,642	462,722	722,402
$n_{voxel} = 2$	7,442	29,282	65,522	116,162	181,202
$n_{voxel} = 3$	3,362	13,122	29,282	51,842	80,802
$n_{voxel} = 4$	1,922	7,442	16,562	29,282	45,602
$n_{voxel} = 5$	1,250	4,802	10,658	18,818	29,282
$n_{voxel} = 6$	882	3,362	7,442	13,122	20,402
$n_{voxel} = 8$	512	1,992	4,232	7,442	11,552
$n_{voxel} = 10$	338	1,250	2,738	4,802	7,442

The number of non-zero elements

Similar to n_{dof} , the number of non-zero elements in the stiffness matrix (n_{nz}) depends on both p , and n_c . By experiment this number of non-zero elements as a function of p and n_{voxel} is obtained, which can be seen in Table 4.2. Contrary to the degrees of freedom, no simple relation exists between n_{nz} , p and n_{voxel} , however, a higher polynomial degree will result in more non-zero indices even when a similar sized stiffness matrix is used.

Table 4.2: The number of non-zero elements (n_{nz}), for different polynomial degrees (p), and different cell topologies. A discretization of 120 by 120 voxels is used

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$	521,284	3,319,204	12,888,004	32,048,404	70,506,964
$n_{voxel} = 2$	131,044	925,444	3,247,204	8,305,924	17,656,804
$n_{voxel} = 3$	58,564	412,164	1,444,804	3,694,084	7,851,204
$n_{voxel} = 4$	33,124	232,324	813,604	2,079,364	4,418,404
$n_{voxel} = 5$	21,316	148,996	521,284	1,331,716	2,829,124
$n_{voxel} = 6$	14,884	103,684	362,404	925,444	1,965,604
$n_{voxel} = 8$	8,464	58,564	204,304	521,284	1,106,704
$n_{voxel} = 10$	5,476	37,636	131,044	334,084	708,964

The half-bandwidth of the stiffness matrix

The shape of the stiffness matrix depends on the numbering scheme, the dimension of the problem, and the polynomial degree p . FCMTOP encompasses the numbering schemes that are present in FCMLab, which are polynomial degree sorting, and topological sorting. For the experiments in this thesis the polynomial degree sorting scheme is used. Two examples of the shape of a stiffness matrix, ordered using this sorting scheme can be seen in Figure 4.1.

The half-bandwidth (n_{bw}) of both matrices is larger than half the size of the stiffness matrix. The numbering scheme sorts \mathbf{D} into the dimensions of the problem. The top half of \mathbf{D} , corresponds to the shape functions in x -direction, while the bottom half corresponds to the shape functions in y -direction. When a 3D example is modeled the bandwidth is thus larger than $2/3 \times n_{dof}$.

After the dimensions are ordered, the numbering scheme divides the coefficients into the corresponding polynomial degrees. The first set of coefficients, corresponding to the x -direction,

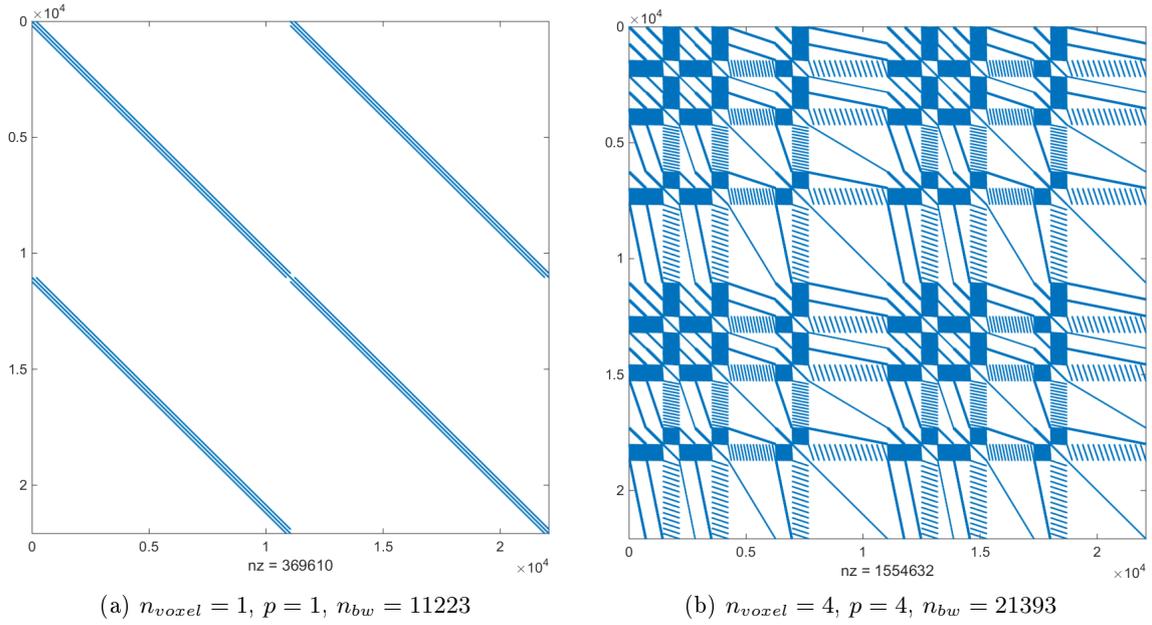


Figure 4.1: Overview of the non-zero indices of stiffness matrix \mathbf{K} , when a polynomial degree sorting scheme is used. The matrix corresponds to the MBB-beam example, where a discretization of 180 by 60 voxels is used

belongs to $p = 1$, the second set of coefficients belongs to $p = 2$, and so on. The topological sorting scheme divides the coefficients into the dimensions as well, and then orders them into the different modes, hence a reduction in bandwidth is not achieved using this numbering scheme.

4.1.3 The effect of p and n_{voxel} on computational cost

A relation has been shown using the memory requirement to store \mathbf{K} , and n_{dof} and n_{nz} . Using the data from Tables 4.1, and 4.2 the memory requirement can be determined as a function of p and n_{voxel} , which is shown in Table 4.3

Table 4.3: The required memory to store the stiffness matrix, for different polynomial degrees (p) and different cell topologies. A discretization of 120 by 120 voxels is used

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{\text{voxel}} = 1$	8.18 MB	51.53 MB	198.64 MB	492.55 MB	1081.36 MB
$n_{\text{voxel}} = 2$	2.06 MB	14.34 MB	50.05 MB	127.62 MB	270.80 MB
$n_{\text{voxel}} = 3$	0.92 MB	6.53 MB	22.27 MB	56.76 MB	120.42 MB
$n_{\text{voxel}} = 4$	0.52 MB	3.60 MB	12.54 MB	31.95 MB	67.77 MB
$n_{\text{voxel}} = 5$	0.33 MB	2.31 MB	8.04 MB	20.46 MB	43.39 MB
$n_{\text{voxel}} = 6$	0.23 MB	1.61 MB	5.59 MB	14.22 MB	30.15 MB
$n_{\text{voxel}} = 8$	0.13 MB	0.91 MB	3.15 MB	8.01 MB	16.98 MB
$n_{\text{voxel}} = 10$	0.09 MB	0.58 MB	2.02 MB	5.13 MB	10.87 MB

It can be seen that the memory requirements are drastically affected by the complexity of the analysis mesh. For $p > n_{\text{voxel}}$, the memory requirement is much larger compared to classical

FEM with linear shape functions, hence these settings are not recommended. Furthermore, it can be seen that even for a small discretization (120 by 120), the memory requirement to store \mathbf{K} is already significant. For large-scale 3D optimization examples, the memory requirement to store \mathbf{K} will become leading. Performing topology optimization on examples with $n_e \geq 10^6$, $p = 2$ and $n_{voxel} = 1$, is not possible on a computer containing 8GB of RAM, due to shortage of memory. More efficient storage algorithms exist to mitigate this problem, however, that is beyond the scope of this thesis.

The computational cost required to obtain the unknown vector \mathbf{D} using Cholesky decomposition, is even more critical than the assembly cost. From Table 4.1 it can be seen that for $p > n_{voxel}$ the size of \mathbf{K} becomes larger than when classical FEM is used, hence these settings are not recommended. What is actually more alarming is that compared to the use of linear shape functions, the half-bandwidth (n_{bw}) is larger when higher polynomial degrees are used. The ordering scheme for the global stiffness matrix dictates n_{bw} , hence a smart ordering scheme can greatly influence the computational efficiency by reducing n_{bw} .

The bandwidth of a matrix can actually be using the reverse Cuthill-McKee algorithm [26]. MATLAB, contains a function to find the ordered indices using the `symrcm` command. Since \mathbf{K} has the same shape at every loop, the reverse Cuthill-McKee algorithm has to be applied only once, before the optimization loop. Using a smart reordering algorithm \mathbf{K} can now be assembled in bandwidth optimized form, without adding extra computational cost. The stiffness matrices shown in Figure 4.1, have been reordered, resulting in the bandwidth optimized matrices shown in Figure 4.2.

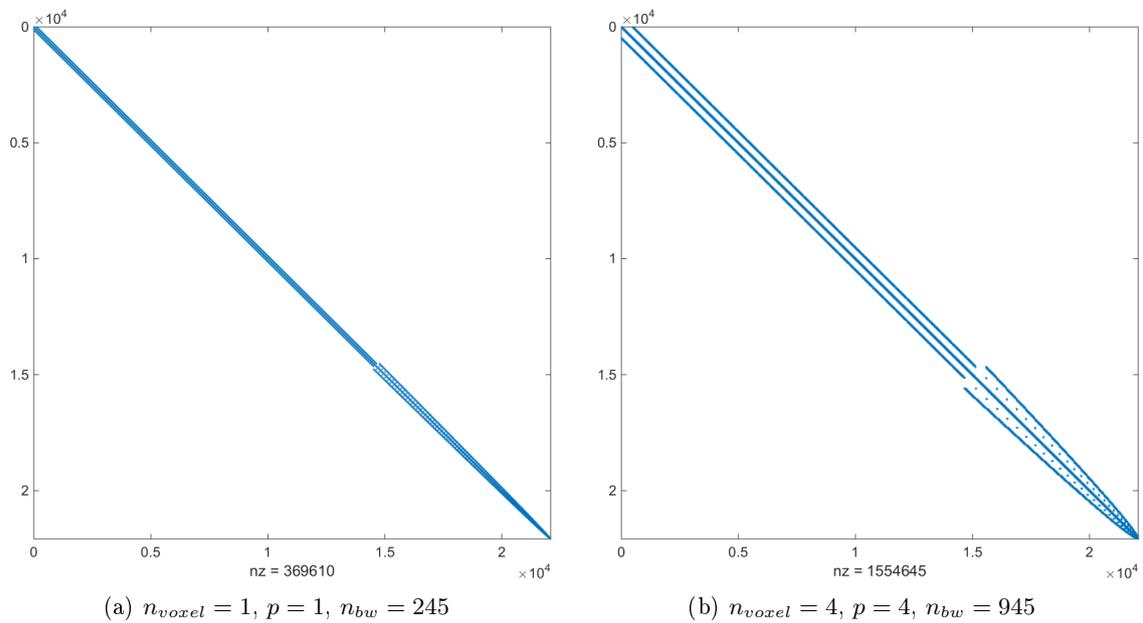


Figure 4.2: Overview of the non-zero indices of stiffness matrix \mathbf{K} , when the reverse Cuthill-McKee algorithm is used to optimize the bandwidth. The matrix corresponds to the MBB-beam example, where a discretization of 180 by 60 voxels is used

4.1.4 Concluding remarks on the computational cost of one design iteration

Based on the relations between p and n_{voxel} , and the computational cost it can be seen that topology optimization cannot be performed efficient when $p > n_{voxel}$. Using these settings, both the memory required for the assembly of \mathbf{K} , and the cost of solving the analysis equations are always larger than when classical FEM is used.

In the next section numerical experiments will be performed to identify for which settings topology optimization using the FCM can be more efficient than topology optimization using classical FEM. Settings for p , and n_{voxel} , with $p > n_{voxel}$, will thus not have to be considered. Furthermore, there are settings for n_{voxel} , and p , which have been identified as unstable. These settings will also not result in a better topology optimization process compared to classical FEM, and can be disregarded as well. The settings that will be tested to identify whether topology optimization using the FCM can be more efficient than topology optimization with classical FEM are indicated by the green cells in Table 4.4.

Table 4.4: The setting that will be used for the numerical experiments on efficiency are indicated by the green cells, the settings marked purple are computationally too expensive, while the red cells correspond to unstable settings

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
$n_{voxel} = 1$					
$n_{voxel} = 2$					
$n_{voxel} = 3$					
$n_{voxel} = 4$					
$n_{voxel} = 5$					
$n_{voxel} = 6$					
$n_{voxel} = 8$					
$n_{voxel} = 10$					

4.2 Experiments on computational efficiency

In the previous section the relations between the computational cost and the parameters n_{voxel} and p have been discussed. In this section the results of numerical experiments will be shown to identify whether these relations hold. For these experiments two large-scale topology optimization examples were used, a 2D Michell truss example using a discretization of 240 by 240 voxels, and a 3D MBB-beam example using a discretization of 120 by 40 by 40 voxels. Both optimization examples were modeled twice, once with bandwidth optimization, and once without, to identify whether bandwidth optimization results in an increase in efficiency. Sensitivity filtering was applied using a filter radius (r_{min}) of 2 voxel widths. The goal of these experiments has been to find the time to assemble the stiffness matrix ($T_{assembly}$) and the time to solve the analysis equations (T_{chol}).

The 2D Michell truss optimization examples have been performed on a HP ZBook 15, with an Intel Core i7-4700MQ processor, 8 GB memory, Windows 8.1, and MATLAB R2014b. Due to memory restrictions the 3D MBB-beam examples have been performed on a Linux cluster using 192 GB RAM. Other processes will be running on both computational systems, making both $T_{assembly}$ and T_{chol} subject to small variations in every design iteration. Despite

these variations, both $T_{assembly}$ and T_{chol} are assumed constant, by taking the mean value of all design iterations. This can be done since the variations are small, and close to the mean value as can be seen in Figure 4.3.

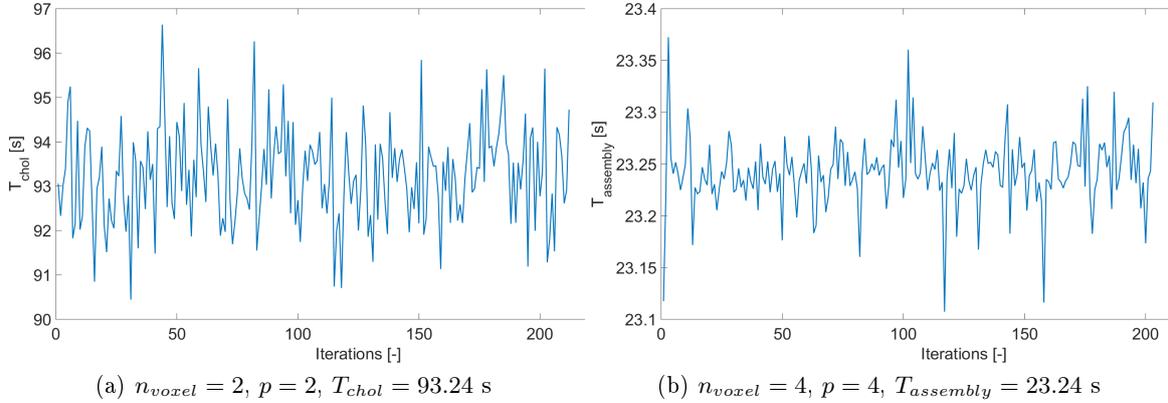


Figure 4.3: Indication of the variations in T_{chol} , and $T_{assembly}$ for every design iteration, in the 3D MBB-beam example

4.2.1 Michell truss

The first example that will be used to test the numerical efficiency of the method is the Michell truss [46]. The Michell truss is widely regarded as the first example of structural optimization, and one of the few examples for which an analytical optimum can be determined. The design domain and its corresponding dimensions are different, and can be seen in the left part of Figure 4.4.

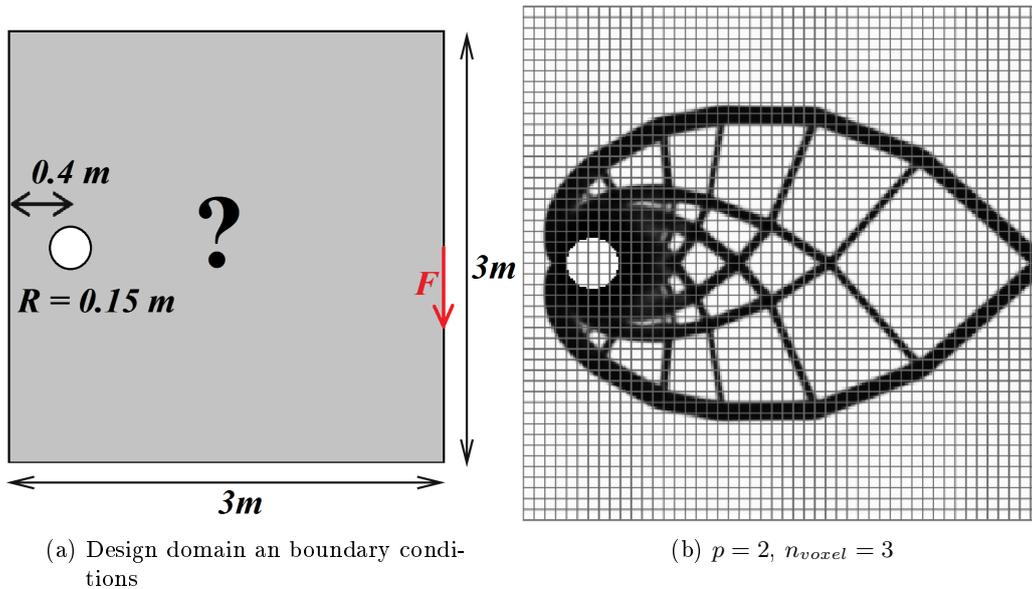


Figure 4.4: The Michell truss topology optimization example, problem formulation, and optimized design

This minimum compliance optimization problem has a similar problem formulation and sensitivity analysis as the MBB-beam example. The material properties of steel are used, $E = 210$ GPa, and $\nu = 0.3$, and the load has a magnitude of 1 MN. Plane stress conditions are assumed with domain thickness $t = 1$ cm. For the optimization a penalization factor $q = 3$, while the maximum allowable volume fraction $V_{max} = 0.2$. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied with a filter radius of 2 voxel widths. The polynomial degree (p), and the number of voxels in each direction of a cell (n_{voxel}), are varied in the experiments.

The experiments have been performed for all settings described in Table 4.4, and it is interesting to note that all optimized density distributions look visually similar to the analytical optimum described by Michell [46]. For the settings $p = 3$, and $n_{voxel} = 5$ the optimized density distribution can be seen in the right part of Figure 4.4. Despite the fact that all experiments produce stable structures, the goal of these experiments is different. The numerical cost of these experiments, described by $T_{assembly}$, T_{chol} , and the total time for one design iteration (T_{loop}) can be seen in Table 4.5.

Table 4.5: The results on numerical cost for different p , and n_{voxel} , using the Michell truss optimization example. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied

p [-]	n_{voxel} [-]	n_{iter} [-]	n_{dof} [-]	n_{bw} [-]	Size \mathbf{K} [MB]	T_{chol} [s]	$T_{assembly}$ [s]	T_{loop} [s]
1	1	358	116162	58323	32.60	0.89	0.42	1.89
1	2	264	29282	14763	8.17	0.18	0.10	0.76
2	2	292	116162	101762	57.24	1.19	0.60	2.45
2	3	287	51842	45442	25.47	0.43	0.25	1.23
2	4	290	29282	25682	14.34	0.22	0.13	0.86
3	3	288	116162	109683	88.92	1.43	0.84	3.07
3	4	281	65522	61863	50.05	0.71	0.46	1.85
3	5	270	42050	39699	32.05	0.43	0.29	1.32
3	6	273	29282	27643	22.27	0.27	0.20	1.06
4	4	270	116162	112503	127.62	1.60	1.08	3.57
4	5	289	74498	72147	81.71	0.91	0.65	2.27
4	6	296	51842	50203	56.76	0.60	0.45	1.67
4	8	373	29282	28353	31.95	0.29	0.24	1.08
5	5	345	116162	113811	173.36	1.82	1.36	4.15
5	6	301	80802	79163	120.42	1.20	0.93	2.96
5	8	294	45602	44673	67.77	0.64	0.50	1.80
5	10	410	29282	28683	43.39	0.39	0.32	1.33

Topology optimization can be performed in a more efficient fashion than for $p = 1$, and $n_{voxel} = 1$. The settings for linear shape functions, combined with $n_{voxel} = 2$, are most efficient for this example, however, for all $n_{voxel} = 2p$, topology optimization can be performed with an increase in computational efficiency compared to classical FEM with linear shape functions.

The time required for Cholesky factorization seems to scale with both n_{bw} , and n_{dof} . A smaller sized stiffness matrix reduces T_{chol} , however the increased half-bandwidth can compensate this reduction in time, as can be seen for $p = 5$, and $n_{voxel} = 6$. Furthermore, the time required to assemble the stiffness matrix, does not completely depend on its memory requirement. A smaller sized stiffness matrix reduces seems to reduce $T_{assembly}$.

The same experiments have been performed, but now bandwidth optimization is performed. The results of these experiments can be seen in Table 4.6.

Table 4.6: The results on numerical cost for different p , and n_{voxel} , using the Michell truss optimization example, including bandwidth optimization. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied

p [-]	n_{voxel} [-]	n_{iter} [-]	n_{dof} [-]	n_{bw} [-]	Size \mathbf{K} [MB]	T_{chol} [s]	$T_{assembly}$ [s]	T_{loop} [s]
1	1	358	116162	963	32.60	1.04	0.42	2.02
1	2	264	29282	483	8.17	0.22	0.11	0.78
2	2	292	116162	1921	57.24	1.52	0.54	2.64
2	3	287	51842	1281	25.47	0.50	0.24	1.27
2	4	290	29282	961	14.34	0.26	0.13	0.90
3	3	292	116162	2875	88.92	1.71	0.75	3.15
3	4	281	65522	2155	50.05	0.82	0.40	1.83
3	5	270	42050	1723	32.05	0.49	0.26	1.31
3	6	273	29282	1435	22.27	0.28	0.18	1.00
4	4	270	116162	3825	127.62	1.35	0.99	3.14
4	5	289	74498	3057	81.71	0.84	0.63	2.13
4	6	296	51842	2545	56.76	0.55	0.44	1.59
4	8	373	29282	1905	31.95	0.26	0.24	1.05
5	5	278	116162	4771	173.36	1.60	1.33	3.84
5	6	301	80802	3971	120.42	1.07	0.92	2.76
5	8	294	45602	2971	67.77	0.56	0.48	1.69
5	10	408	29282	2371	43.39	0.35	0.32	1.29

It is very interesting to observe that the bandwidth optimization, does not necessarily lead to a more efficient Cholesky decomposition. Only when $p = 4$ or $p = 5$ are used the bandwidth optimization result in a slightly more efficient optimization. The reason for this is still unknown. Furthermore, it can be noted that some experiments show a difference in the number of iterations between the bandwidth optimized case, and the non-bandwidth optimized case. The reason can be a small difference in \mathbf{D} due to different conditioning errors after reordering \mathbf{K} . This difference can then accumulate over the number of iterations, hence a slightly different optimization is performed.

4.2.2 3D MBB-Beam example

The second optimization example that is considered is the 3D MBB-beam example, with a discretization of 120 by 40 by 40 voxels. The boundary conditions of this optimization example are almost similar to the boundary conditions and the dimensions shown for the 2D MBB-beam example shown in Figure 2.5. The difference is the third dimension, where the domain has a length of 1 m in the z -direction. The edge BC becomes a face BC, and the nodal BCs become edge BCs. A volume fraction (V_{max}) of 0.2 is used, while all other properties are similar to the 2D MBB-beam example.

The optimization experiments have been performed for all settings described in Table 4.4. The numerical cost of these experiments, described by $T_{assembly}$, T_{chol} and (T_{loop}) can be seen in Table 4.7. Similar to the Michell truss example a valid density distribution has been achieved for all settings. The optimized density distribution for the settings $p = 2$ and $n_{voxel} = 4$ can be seen in Figure 4.5.

Table 4.7: The results on numerical cost for different p , and n_{voxel} , using the 3D MBB-beam optimization example. A discretization of 120 by 40 by 40 voxels is used, and sensitivity filtering is applied

p [-]	n_{voxel} [-]	n_{iter} [-]	n_{dof} [-]	Size \mathbf{K} [MB]	T_{chol} [s]	$T_{assembly}$ [s]	T_{loop} [s]
1	1	223	610203	730.49	133.04	11.34	149.36
1	2	224	80703	93.11	2.96	1.29	6.88
2	2	212	610203	1716.87	93.24	16.10	118.60
2	4	299	80703	217.76	3.24	1.85	8.03
3	4	199	262353	1414.19	16.40	10.46	33.69
3	5	226	136875	726.88	6.29	5.56	16.38
4	4	208	610203	5755.50	85.38	46.04	152.28
4	5	203	316899	2953.99	30.63	23.24	65.50
4	8	235	80703	726.45	8.99	5.51	19.65
5	5	223	610203	9124.16	98.72	75.32	205.37
5	8	219	154128	2238.71	26.82	19.63	56.92
5	10	234	80703	1150.05	6.15	9.90	22.04

The cost of performing an optimization $p = 1$, and $n_{voxel} = 1$ is higher, compared to the cost for $p = 2$ and $n_{voxel} = 2$. This is not expected due to the lower bandwidth and fewer non-zero elements. The difference cannot be explained, but it is assumed that another program running on the same node might have caused this difference. Apart from this it can be seen that a separate density and analysis mesh can reduce the optimization time by a factor 10! This very large difference, shows the benefits of performing topology optimization using the FCM.

The settings for linear shape functions combined with $n_{voxel} = 2$, are the most efficient. For all $n_{voxel} = 2p$, topology optimization can be performed with an increase in computational efficiency compared to classical FEM with linear shape functions. Furthermore, it can be seen that the required memory to store \mathbf{K} drastically increases when higher-order shape functions are used. In 3D examples the amount of introduced degrees of freedom grows at a much faster rate compared to 2D analysis. Therefore, its important to consider the memory requirement to store \mathbf{K} in 3D topology optimization applications, using higher-order shape functions.

Similar to the Michell truss bandwidth optimization is applied, however, in this case that did not result in a more efficient optimization, hence the corresponding results will not be shown here.

4.3 Concluding remarks

In this chapter the relations, between the size and shape of the stiffness matrix, and the computational cost of performing topology optimization has been discussed. A relation between p , n_{voxel} and the time required to solve the analysis equations has been identified, where the higher-order shape functions increase the bandwidth, and thus the corresponding computational cost. To reduce the computational cost bandwidth optimization has been implemented using the reverse Cuthill-McKee algorithm, however, this did not seem to reduce the analysis time, sometimes the analysis time even increased. A reason for this might be that the function to perform Cholesky decomposition in MATLAB uses a more efficient re-ordering scheme.

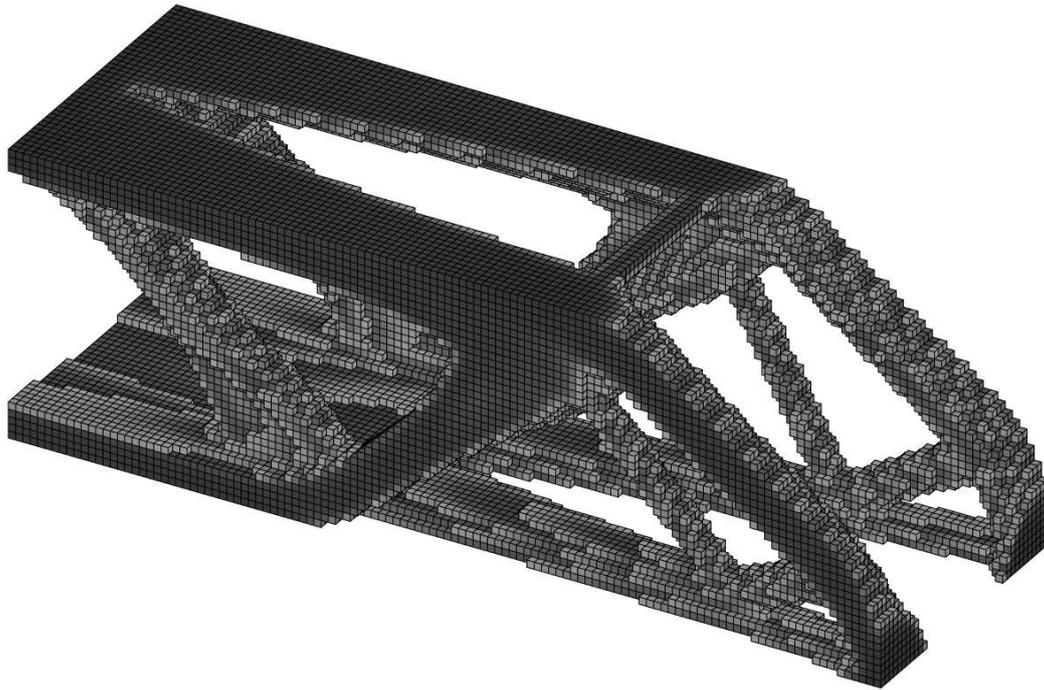


Figure 4.5: The optimized density distribution for the 3D MBB-beam optimization example, with $p = 2$ and $n_{voxel} = 4$. A discretization of 120 by 40 by 40 voxels is used, and sensitivity filtering is applied

Furthermore, a relation between p , n_{voxel} and the required memory to store the stiffness matrix has been identified. The use of higher-order shape functions introduces more non-zero elements in the stiffness matrix, thus increases the required memory to store \mathbf{K} . Especially in 3D optimization examples this memory requirement can become a limiting factor for the size of optimization problems that can be modeled.

The most important conclusion that can be drawn from the experiments in this chapter, is that topology optimization using a separate density and analysis mesh can greatly reduce the computational cost. In 2D examples an increase in numerical efficiency of a factor 2 can be identified, without compromising the optimization objective. In 3D topology optimization examples the computational cost can be decreased by more than a factor 10.

Discussion on the potential of topology optimization using the FCM

In the previous chapters, the stability, accuracy, and efficiency of performing topology optimization using the FCM have been identified. In this chapter the performance of topology optimization using the FCM is compared to the performance of topology optimization using classical FEM with linear shape functions. To find when the use of the FCM is more beneficial than classical FEM in topology optimization applications.

5.1 The performance of topology optimization using the FCM

In this section the results of the previous two chapters will be summarized. Topology optimization examples with a larger length-scale will be shown to emphasize the potential of using a separate density and analysis mesh. Furthermore, the use of higher-order shape functions will be discussed.

5.1.1 The reliability, efficiency and accuracy experiments

From the experiments presented in the previous chapters it can be seen that topology optimization using the FCM can be performed in a stable fashion. The need for a length-scale imposed using mesh-independent filtering is shown, but, this length-scale is small and does not violate the required high resolution. Furthermore, the settings are shown for p and n_{voxel} for which topology optimization can be performed in a stable fashion. More importantly, it is shown that these different analysis systems may all result in a different local optimum, however, the corresponding differences in post-processed objectives are small. It is very important that this post-processing of the objective is done. It may show artificially stiff cells, which otherwise would have been overlooked. Furthermore, it allows for a fair comparison between the different settings of p , and n_{voxel} .

More interesting to note is that the use of a decoupled density and analysis mesh can greatly reduce the computational cost of topology optimization, without compromising the optimization objective. Especially in 3D topology optimization examples the computational cost can be decreased by more than a factor 10. The 3D experiments are performed on a cluster, which may lead to variations in the time measurements. Nevertheless, the benefit of a separate density and analysis mesh is shown. In the 2D optimization example large reductions in the optimization time can be seen as well.

So far, no setting for p , and n_{voxel} can be identified as being the best setting. A higher-order shape function does not seem to be mandatory when multi-resolution topology optimization is performed. This observation will be discussed in more detail later in this chapter. First, a further reduction in computational cost using a larger length-scale will be discussed.

5.1.2 Topology optimization examples with a larger length-scale

High-resolution large-scale topology optimization can only be performed using a small filter radius (r_{min}). A fine discretization combined with a large filter radius, will make sure the boundary of structural members becomes smooth, however, the size of these members needs to have a minimum length-scale, hence a truly high-resolution cannot be achieved. Nevertheless, most optimization examples use a mesh-independent filter radius, which is larger than 2 voxel widths.

This larger length-scale, means that the analysis mesh can become less complex, hence the computational cost can be decreased. To demonstrate this, consider the compliant force inverter example, with a fine discretization of 240 by 120. Robust topology optimization is applied with a filter radius of 4 voxel widths. The corresponding optimization time per loop T_{loop} , for different settings of p , and n_{voxel} can be seen in Table 5.1.

Table 5.1: The results on numerical cost for different p , and n_{voxel} , using the compliant force inverter example. A discretization of 240 by 120 voxels is used, robust topology optimization is applied ($\eta = 0.2$), with $r_{min} = 4$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	T_{loop} [s]	u_{out} [cm]
1	1	58322	5.10	-1.64
1	4	3782	0.94	-1.82
1	5	2450	0.97	-2.13
2	8	3782	1.02	-2.19
2	10	2450	0.96	-2.21
2	12	1722	0.86	-1.86
3	10	5402	1.14	-2.18
3	12	3782	1.06	-2.13
3	15	2450	0.90	-1.79
4	12	6642	1.22	-1.82
4	15	4290	1.01	-1.60
4	20	2450	0.96	-2.08
5	15	6642	1.49	-1.77
5	20	3782	1.09	-1.90

Compared to the filter radius of 2 voxel widths the computational efficiency can be reduced even more. It is very interesting to see that the time spent in the optimization loop can be reduced more than five times compared to the use of classical FEM with linear shape

functions. This means that the additional cost for robust topology optimization using $p = 1$ and $n_{voxel} = 1$ can be alleviated using a separated density and analysis mesh.

Contrary to the previous experiments, there is no setting which is clearly more efficient than another one. The quality of the analysis mesh also does not seem to affect the output displacement. The variation in this output displacement seems random, and is just the result of optimizations getting stuck at local minima. The highest output displacement -2.21 cm, is achieved using a very simple analysis mesh $p = 2$ and $n_{voxel} = 10$, so no highly accurate displacement field is required to solve this type of problems.

5.1.3 The use of higher-order shape functions

Almost all experiments performed so far, have shown that topology optimization using $p = 1$ and $n_{voxel} > 1$ is the most, or among the most efficient settings to perform computationally efficient topology optimization. This polynomial degree results in the most diagonal stiffness matrix, hence the cost for solving the analysis equations is low. Nguyen et al [47, 48] already have proposed a multi-resolution approach using linear shape functions, similar to topology optimization using the FCM with $p = 1$. This raises the question whether higher-order shape functions should be used at all.

The optimized density distributions using $p = 1$ are similar to the density distributions obtained using higher-order shape functions. After post-processing the corresponding values for the compliance or output displacement are similar as well. It seems that the optimization problems discussed so far, do not require a highly accurate displacement field to perform topology optimization. An error in the displacement field is not bad, as long as this error is approximately similar for the whole analysis system. This seems the case, when a simple analysis mesh is used with $p = 1$, and $n_{voxel} = 2$. This means that multi-resolution topology optimization using $p = 1$ is very well suited for relatively simple topology optimization problems. The simple analysis mesh will greatly reduce the computational cost of performing topology optimization. The amount of voxels per cell can be scaled up with the applied length-scale to prevent the occurrence of the artificial stiff cells, yet reducing the computational cost.

It is very important to remember that topology optimization problems, where the boundary is enforced in the weak sense, require an accurate displacement field. Recall the topology optimization example of the Michell truss, where the circular boundary has been applied in the weak sense. The optimized density distributions in this example are all regarded as stable, however, the differences between the objective and the post-processed compliances are very large as can be seen in Table 5.2.

The values for the post-processed compliance (C) are almost similar, while differences in the objectives are alarming. The reason is the application of the weak Dirichlet BC, which requires an integration over a parameterized boundary using the shape functions present in the analysis system. For a very simple analysis system, the BCs are therefore not applied correctly. In the Michell truss example, this did not cause large problems, since only one BC was used. The corresponding stiffness of the boundary is estimated wrong, but this error is constant in the whole analysis mesh, hence the sensitivity analysis is still correct. But if a strong BC is added to the system, the corresponding analysis would be completely wrong.

An optimization problem with multiple weak BCs therefore requires a much more complex analysis system, than an optimization problem with strong BCs. This is where the higher-order shape functions can be beneficial over linear FEM, since these higher-order functions are able to describe a more complex displacement field for similar n_{dof} [58, 72]. A detailed

Table 5.2: The objective and post-processed compliance, for different p , and n_{voxel} , using the Michell truss optimization example. A discretization of 240 by 240 voxels is used, and sensitivity filtering is applied

p [-]	n_{voxel} [-]	n_{iter} [-]	n_{dof} [-]	$\hat{F}(\boldsymbol{\rho})$ [-]	C [J]
1	1	358	116162	1.80E+09	9.05E+04
1	2	264	29282	1.17E+05	8.49E+04
2	2	292	116162	8.41E+04	8.62E+04
2	3	287	51842	1.02E+05	8.64E+04
2	4	290	29282	7.64E+04	8.56E+04
3	3	288	116162	1.80E+05	8.68E+04
3	4	281	65522	3.23E+05	8.56E+04
3	5	270	42050	8.69E+04	8.56E+04
3	6	273	29282	8.44E+04	8.62E+04
4	4	270	116162	2.40E+05	8.55E+04
4	5	289	74498	7.85E+04	8.52E+04
4	6	296	51842	7.98E+04	8.53E+04
4	8	373	29282	7.64E+04	8.55E+04
5	5	345	116162	8.65E+04	8.57E+04
5	6	301	80802	8.13E+04	8.58E+04
5	8	294	45602	7.77E+04	8.52E+04
5	10	410	29282	1.75E+05	8.60E+04

study into which settings of p and n_{voxel} should be used for correct enforcement of the BC is not performed in this thesis. However, it is strongly recommended to perform this study in the future.

Other topology optimization problems that require an accurate representation of the displacement field, are optimization problems with displacement constraints, problems with non-linear strain, but most importantly optimization problems with a stress-constraint. As the name implicates stress-constrained topology optimization is constrained by a maximum allowable stress [42]. The higher-order shape functions are much better in describing an accurate stress field than lower order shape functions. Unfortunately, the post-processor in FCMTOP is not able to show these highly accurate stress distribution for different p and n_{voxel} . Therefore the potential of higher-order shape functions for stress-constrained optimization cannot be demonstrated.

To conclude this discussion it is important to consider that the use of higher-order shape functions completely depends on the type of optimization problem. For relatively simple optimization problems the use of a relatively low polynomial degree $p = 1, 2, 3$ is sufficient to perform very efficient topology optimization using a separate density and analysis mesh. For more difficult topology optimization examples a more accurate analysis system is required. For both types of problems the FCM is well suited. Due to its general applicability this method can be used in a wide range of problems, where the settings can be tweaked for every problem to keep the computational cost as low as possible.

Finally, it has to be mentioned that the computational cost of topology optimization using higher-order shape functions can be further reduced using static condensation of the stiffness matrix.

Reducing the degrees of freedom using static condensation

A high polynomial degree introduces a large number of internal modes and corresponding coefficients, especially in the product space, which is used in FCMTOP. These higher order internal modes are local to the element and can be condensed out of the stiffness matrix, drastically reducing its size. Several studies have shown that this does not affect the accuracy of the displacement field, and even can lower the condition number of the stiffness matrix [44, 71].

Static condensation has not been applied in FCMTOP, however, the use of it may have a significant impact on the computational cost using a high p . It is strongly recommended that this method should be implemented in the future, and tested on various topology optimization examples.

5.2 Large-scale topology optimization using FCMTOP

Large-scale topology optimization examples requires the use of a separate density and analysis mesh to keep the computational cost low. When multiple voxels per cell are used, a smart way of boundary enforcement has to be considered, for which higher-order shape functions are best suited. Hence, the FCM is well suited to be used for large-scale topology optimization applications. To demonstrate this, a large scale minimum compliance optimization of a clamped wing will be shown with a discretization of 200 by 40 by 400 voxels.

For the shape of the wing a NACA 0020 airfoil is used, with a chord length of 1 m, and a wingspan 2.2 m. The corresponding domain dimensions are 1.1 by 0.44 by 2.2 m, with $V_{max} = 0.25$. The outer shape of the wing is modeled in FCMTOP using passive elements. The aerodynamic load will be applied on the wing surface using weak enforcement of the Neumann boundaries. The discussion in the previous section indicated that a high p is preferred in case weak BCs are applied. To do this a polynomial degree of 4 is used, with $n_{voxel} = 8$. The wing is made up of aluminum 7075-T6, which is commonly used in aerospace applications, $E = 71.7$ GPa and $\nu = 0.33$. Density filtering with modified Heaviside projection is applied for $r_{min} = 2.2$ cm, to achieve a high-resolution design.

The aerodynamic loading on the wing is calculated using XFOIL, which is a panel code developed by Drela [29]. A viscous analysis is performed at sea-level conditions ($\rho_{air} = 1.225$ kg/m³, $T = 291.25$ K). The wing is flying at $Mach = 0.2$, at an angle of attack ($\alpha = 5^\circ$). In Figure 5.1 a cross-section of the wing is shown as well as the corresponding pressure distribution. The lines pointing into the wing correspond to a pressure compressive force at the surface, while the lines pointing outward correspond to a tensile force.

This pressure distribution is applied at ten span-wise positions of the wing. At these positions the pressure is integrated, over 140 different parameterized lines that form the outer shape of the airfoil. The elements outside of the airfoil are set to be passive with no density at all, and the wing is clamped at the root. The corresponding optimized wing can be seen in Figure 5.2, where four different cross-sections are shown to give an overview of the internal structure of the wing.

It is very interesting to see that the optimized density distribution provides bending-stiffness to the wing, in a similar fashion as in done by using a wing-box. The optimizer goes to an optimum which is known to be the best, and applied in almost all structural designs of wings. Close to the root the bending moment caused by the pressure load is the largest,

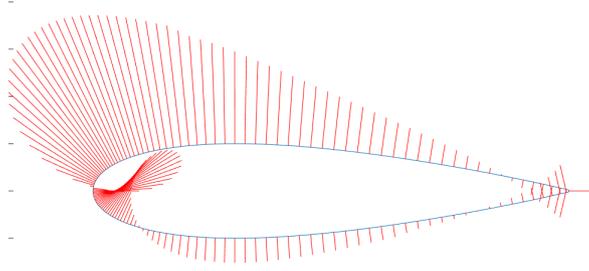


Figure 5.1: The shape of a NACA 0020 airfoil, and the corresponding pressure distribution obtained in XFOIL using a viscous analysis is performed at sea-level conditions ($\rho_{air} = 1.225 \text{ kg/m}^3$, $T = 291.25 \text{ K}$), $Mach = 0.2$, and ($\alpha = 5^\circ$)

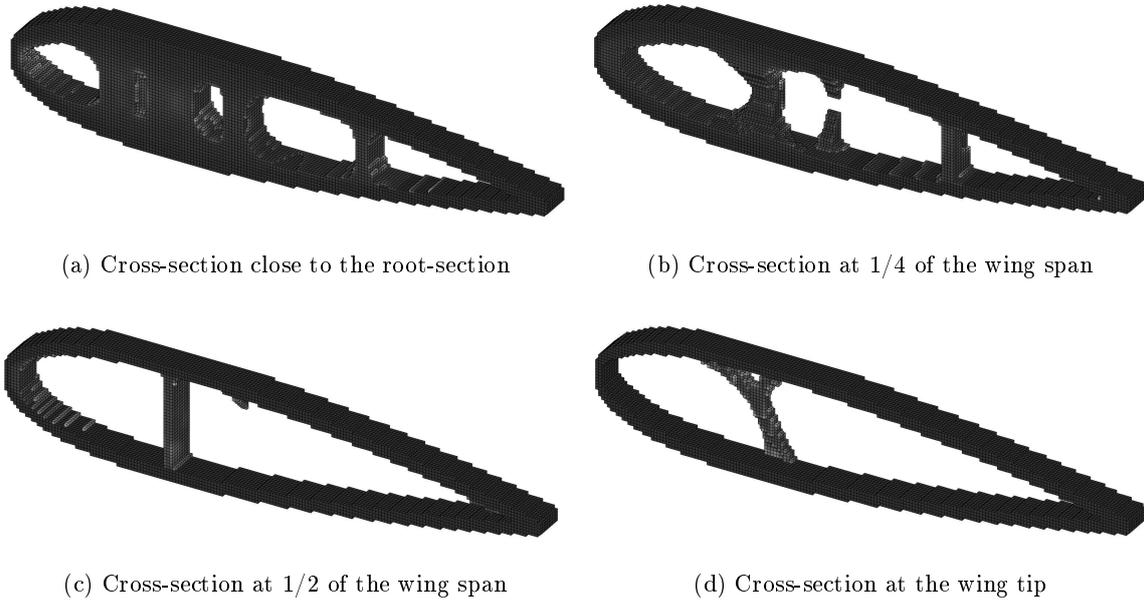


Figure 5.2: Different cross-sections of the clamped wing example. A discretization of 200 by 40 by 400 voxels is used, density filtering with modified Heaviside projection is applied, with $r_{min} = 2$ voxel widths

hence the structural members are thickest. Close to the tip, the corresponding stresses are small resulting in a thin structure.

This optimization example shows the great potential of topology optimization. Despite the fact that the density distribution is very coarse, the optimizer is able to create a structure that can be recognized to one used in engineering practice. The 3.2 million design elements used in this example, are close to the maximum that FCMTOP can handle. Even with $p = 4$, and $n_{voxel} = 8$, the optimization process lasted more than 2 days. If $p = 1$ and $n_{voxel} = 1$ would have been used, this optimization would not have been possible, hence a separated density and analysis mesh is extremely beneficial when large-scale topology optimization is considered.

Chapter 6

Final remarks

6.1 Conclusions

The research objective of this thesis was: *"To investigate the potential of the FCM for large scale, high resolution topology optimization problems, compared to the use of finite elements (FE) with linear shape functions in terms of reliability, efficiency and accuracy."* The first step of reaching this objective has been to develop a method to perform topology optimization using the FCM. To fully exploit the properties of the FCM, density based optimization is performed using the SIMP method for material interpolation. The theory used to develop this model, called the Finite Cell Method Topology Optimization Program (FCMTOP) is discussed in Chapter 2.

In Chapter 3 the need for a length-scale imposed using mesh-independent filtering is shown. This prevents the formation of artificially stiff cells, in a similar fashion as a checkerboard pattern is prevented. Furthermore, values for the polynomial degree p , and the number of voxels in each cell direction n_{voxel} , are shown that ensure stable topology optimization. It was very interesting to see that these different settings all produced visually similar looking topologies. The different analysis systems led to different values for the objectives, however, when all results are post-processed using a highly accurate mesh the corresponding objectives are very similar.

The use of a decoupled density and analysis mesh can greatly reduce the computational cost of topology optimization as has been discussed in Chapter 4. Especially in 3D topology optimization examples the computational cost can be decreased by more than a factor 10, while maintaining a high-resolution in the density field. The use of a larger length-scale can reduce the computational cost even more, which is especially beneficial for robust topology optimization.

It is important to consider that the use of higher-order shape functions completely depends on the type of optimization problem. For relatively simple optimization problems the use of a relatively low polynomial degree $p = 1, 2, 3$ is sufficient to perform very efficient topology optimization. For more difficult topology optimization examples, such as stress-constrained topology optimization, or examples with multiple weak boundary conditions, a more accurate analysis system is required, hence a larger polynomial degree should be used.

To conclude, the FCM is very well suited to adapt its analysis system to the complexity of the topology optimization problem. The experiments done so far, show the superiority of a separate density and analysis mesh, over topology optimization using classical FEM with linear shape functions. The use of the FCM is therefore strongly recommended to keep the computational cost of topology optimization low.

6.2 Recommendations

- When topology optimization is performed with a separate density and analysis mesh, it is very important to realize that a relatively simple analysis method is used. To make sure that the optimized density distribution is really optimal, post-processing with a more advanced analysis system has to be performed. This post-processing may show artificially stiff cells, which otherwise would have been overlooked. Furthermore, it allows for a fair comparison between the different settings of p , and n_{voxel} .
- So far, optimization problems using relatively simple boundary conditions are considered. The enforcement of weak boundary conditions require a highly accurate displacement field, hence the use of higher order shape functions is required. A study into which settings of p and n_{voxel} should be used for correct enforcement has to be performed. Furthermore, this study should include a comparison on the computational cost of this process with respect to classical FEM with linear shape functions.
- The use of higher-order shape functions, and a separate density and analysis mesh is at the moment not considered for stress-constrained topology optimization. The FCM shows great potential for this, since it can not only reduce the computational cost, but also increase the accuracy of the stress field. A research study to identify this potential is therefore highly recommended.
- A high polynomial degree introduces a large number of internal modes and corresponding coefficients. These higher order internal modes are local to the element and can be condensed out of the stiffness matrix, drastically reducing its size. It is recommended that this method is applied in FCMTOP to identify its effect on the accuracy, stability, and numerical efficiency when used in combination with topology optimization.
- FCMTOP is programmed in MATLAB, and is therefore not very efficient. The developed topology optimization could therefore be programmed in a more efficient environment such as PETSc [1], to identify its capabilities in very large-scale topology optimization examples.

References

- [1] N. Aage, E. Andreassen, and B.S. Lazarov. Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, 51:565–572, 2014.
- [2] N. Aage and B.S. Lazarov. Parallel framework for topology optimization using the methods of moving asymptotes. *Structural and Multidisciplinary Optimization*, 47:493–505, 2013.
- [3] N. Aage, T.H. Poulsen, A. Gersbors-Hansen, and O. Sigmund. Topology optimization of large scale stokes flow problems. *Structural and Multidisciplinary Optimization*, 35:175–180, 2008.
- [4] G. Allaire, F. Jouve, and A.M. Toader. A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334:1125–1130, 2002.
- [5] G.M. Amdahl. Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS conference proceedings*, volume 30, pages 483–485, 1967.
- [6] O. Amir, N. Aage, and B.S. Lazarov. On multigrid-cg for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49:815–829, 2014.
- [7] O. Amir, M.P. Bendsøe, and O. Sigmund. Approximate reanalysis in topology optimization. *International Journal for Numerical Methods in Engineering*, 78:1474–1491, 2009.
- [8] O. Amir and O. Sigmund. On reducing computational effort in topology optimization: how far can we go? *Structural and Multidisciplinary Optimization*, 44:25–29, 2011.
- [9] O. Amir, M. Stolpe, and O. Sigmund. Efficient use of iterative solvers in nested topology optimization. *Structural and Multidisciplinary Optimization*, 42:55–72, 2010.
- [10] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16, 2011.

-
- [11] T. Ansola, E. Veguería, J. Canales, and J. Tárrago. A simple evolutionary topology optimization procedure for compliant mechanism design. *Finite Elements in Analysis and Design*, 44:53–62, 2007.
- [12] T. Ansola, E. Veguería, A. Maturana, and J. Canales. 3d compliant mechanisms synthesis by a finite element addition procedure. *Finite Elements in Analysis and Design*, 46:760–769, 2010.
- [13] I. Babuška, B.A. Szabó, and I.N. Katz. The p-version of the finite element method. *SIAM Journal on Numerical Analysis*, 18:515–545, 1981.
- [14] M.P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.
- [15] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–242, 1988.
- [16] M.P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69:635–654, 1999.
- [17] M.P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer-Verlag, Berlin, Germany, 2003.
- [18] T. Borvall and J. Petersson. Large-scale topology optimization in 3d using parallel computing. *Computer Methods in Applied Mechanics and Engineering*, 190:6201–6229, 2001.
- [19] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50:2143–2158, 2001.
- [20] B. Bourdin and A. Chambolle. Design-dependent loads in topology optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 9:19–48, 2003.
- [21] T.E. Bruns. A reevaluation of the simp method with filtering and an alternative formulation for solid-void topology optimization. *Structural and Multidisciplinary Optimization*, 30:428–436, 2005.
- [22] T.E. Bruns and D.A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190:3443–3459, 2001.
- [23] P.W. Christensen and A.Klarbring. *An Introduction to Structural Optimization*. Springer, Linköping, Sweden, 2009.
- [24] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2002.
- [25] J.A. Cottrel, T.J.R Hughes, and Y. Bazilevs. *Isogeometric Analysis: Towards Integration of CAD and FEA*. John Wiley & Sons, Ltd., West Sussex, United Kingdom, 2009.
- [26] T.A. Davies. *Direct Methods for Sparse Linear Systems*. Society of Industrial and Applied Mathematics, Philadelphia, United States of America, 2006.

- [27] J.D. Deaton and R.V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49:1–38, 2013.
- [28] A. Diaz and O. Sigmund. Checkerboard patterns in layout optimization. *Structural Optimization*, 10:40–45, 1995.
- [29] M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In Low Reynolds Number Aerodynamics, editor, *T.J. Mueller*, volume 54 of *Lecture Notes in Engineering*, pages 1–12. Springer Berlin Heidelberg, 1989.
- [30] A. Düster, H. Bröker, and E. Rank. The p-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering*, 52:673–703, 2001.
- [31] A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197:3768–3782, 2008.
- [32] H.A. Eschenauer, V.V. Koblelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural and Multidisciplinary Optimization*, 8:42–51, 1994.
- [33] A. Evgrafov, C.J. Rupp, K. Maute, and M.L. Dunn. Large-scale parallel topology optimization using a dual-primal substructuring solver. *Structural and Multidisciplinary Optimization*, 36:329–345, 2008.
- [34] J.K. Guest and L.C. Smith Genut. Reducing dimensionality in topology optimization using adaptive design variable fields. *International Journal for Numerical Methods in Engineering*, 81.
- [35] J.K. Guest, J.H. Prévost, and T. Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection filters. *International Journal for Numerical Methods in Engineering*, 61:238–254, 2004.
- [36] X. Huang and Y.M. Xie. A further review of eso type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 41:671–683, 2010.
- [37] I.Y. Kim and B.M. Kwak. Design space optimization using a numerical design continuation method. *International Journal for Numerical Methods in Engineering*, 53:1979–2002, 2002.
- [38] T.S. Kim, J.E. Kim, and Y.Y. Kim. Parallelized structural topology optimization for eigenvalue problems. *International Journal of Solids and Structures*, 41:2623–2641, 2004.
- [39] Y. Y. Kim and G. H. Yoon. Multi-resolution multi-scale topology optimization- a new paradigm. *International Journal of Solids and Structures*, 37:5529–5559, 2000.
- [40] L. Krog, A. Tucker, and G. Rollema. Application of topology, sizing and shape optimization methods to optimal design of aircraft components. In *Proc. 3rd Altair UK HyperWorks Users Conference*, 2002.
- [41] B.S. Lazarov and O. Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86:765–781, 2011.

- [42] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41:605–620, 2010.
- [43] A. Mahdavi, R. Balaji, M. Frecker, and E.M. Mockensturm. Topology optimization of 2d continua for minimum compliance using parallel computing. *Structural and Multidisciplinary Optimization*, 32:121–132, 2006.
- [44] J. Mandel. Iterative solvers by substructuring for the p-version finite element method. *Computer Methods in Applied Mechanics and Engineering*, 80:117–128, 1990.
- [45] T.H.G. Megson. *Aircraft Structures for Engineering Students*. Elsevier, 2012.
- [46] A.G.M. Michell. The limits of economy of material in frame structures. *Philosophical Magazine*, 8:589–597, 1904.
- [47] T.H. Nguyen, G.H. Paulino, J. Song, and C.H. Le. A computational paradigm for multiresolution topology optimization (mtop). *Structural and Multidisciplinary Optimization*, 41:525–539, 2010.
- [48] T.H. Nguyen, G.H. Paulino, J. Song, and C.H. Le. Improving multiresolution topology optimization via multiple discretizations. *International Journal for Numerical Methods in Engineering*, 92:507–530, 2012.
- [49] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen Aus Dem Mathematischen Seminar Der Universität Hamburg*, 36(1):9–15, 1971.
- [50] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [51] S.J. Osher and F. Santosa. Level set methods for optimization problems involving geometry and constraints. *Journal of Computational Physics*, 171:272–288, 2001.
- [52] J. Parvizian, A. Düster, and E. Rank. Finite cell method: h- and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics*, 41:121–133, 2007.
- [53] J. Parvizian, A. Düster, and E. Rank. Topology optimization using the finite cell method. *Optimization and Engineering*, 13:57–78, 2012.
- [54] T.A. Poulsen. Topology optimization in wavelet space. *International Journal for Numerical Methods in Engineering*, 53:567–582, 2002.
- [55] O.M. Querin, G.P. Steven, and Y.M. Xie. Evolutionary structural optimisation (eso) using a bidirectional algorithm. *Engineering Computations*, 15:1031–1048, 1998.
- [56] G.I.N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37:217–237, 2009.
- [57] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank. The finite cell method for bone simulations: verification and validation. *Biomechanics and Modeling in Mechanobiology*, 11:425–437, 2012.

- [58] D. Schillinger and M. Ruess. The finite cell method: A review in the context of higher-order structural analysis of cad and image-based geometric models. *Archives of Computational Methods in Engineering*, 2014.
- [59] O. Sigmund. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines: An International Journal*, 25:493–524, 1997.
- [60] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.
- [61] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33:401–424, 2007.
- [62] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 25:227–239, 2009.
- [63] O. Sigmund and K. Maute. Sensitivity filtering from a continuum mechanics perspective. *Structural and Multidisciplinary Optimization*, 46:471–475, 2012.
- [64] O. Sigmund and K. Maute. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48:1031–1055, 2013.
- [65] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16:68–75, 1998.
- [66] R. Stainko. An adaptive multilevel approach to the minimal compliance problem in topology optimization. *Communications in Numerical Methods in Engineering*, 22:109–118, 2006.
- [67] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance in topology optimization. *Structural and Multidisciplinary Optimization*, 22:116–124, 2001.
- [68] K. Suresh. Efficient generation of large-scale pareto-optimal topologies. *Structural and Multidisciplinary Optimization*, 47:49–61, 2013.
- [69] K. Svanberg. The method of moving asymptotes: A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
- [70] I.Y. Kim S.Y. Kim and C.K. Mechefske. A new efficient convergence criterion for reducing computational expense in topology optimization: reducible design variable method. *International Journal for Numerical Methods in Engineering*, 90:752–783, 2012.
- [71] B. Szabó and I. Babuška. *Finite Element Analysis*. John Wiley & Sons, Inc., Hoboken, New Jersey, 1991.
- [72] B. Szabó, A. Düster, and E. Rank. The p-version of the finite element method. In Encyclopedia of computational mechanics, editor, *E. Stein and R. de Borst and T.J.R. Hughes*, volume 1, chapter 5, pages 119–139. 2004.
- [73] A. Takezawa, S. Nishiwaki, and M. Kitamura. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 229:2697–2718, 2010.

- [74] N.P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48:437–472, 2013.
- [75] K. Vemaganti and W.E. Lawrence. Parallel methods for optimality criteria-based topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 194:3639–3667, 2005.
- [76] S. Venkataraman and R.T. Haftka. Structural optimization complexity: what has moore’s law done for us? *Structural and Multidisciplinary Optimization*, 28:375–287, 2004.
- [77] E. Wadbro and M. Berggren. Megapixel topology optimization on a graphics processing unit. *SIAM Review*, 51:707–721, 2009.
- [78] F. Wang, B.S. Lazarov, and O. Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43:767–784, 2011.
- [79] M.Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192:227–246, 2003.
- [80] S. Wang, E. de Sturler, and G.H. Paulino. Large-scale topology optimization using preconditioned krylov subspace method with recycling. *International Journal for Numerical Methods in Engineering*, 69:2441–2468, 2007.
- [81] Y.M. Xie and G.P. Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49:885–896, 1993.
- [82] Y.M. Xie and G.P. Steven. *Evolutionary Structural Optimization*. Springer-Verlag, London, United Kingdom, 1997.
- [83] Z. Yang, M. Ruess, S. Kollmannsberger, A. Düster, and E. Rank. An efficient integration technique for the voxel-based finite cell method. *International Journal for Numerical Methods in Engineering*, 91:457–471, 2012.
- [84] V. Young, O.M. Querin, G.P. Steven, and Y.M. Xie. 3d and multiple load case bi-directional evolutionary structural optimization (beso). *Structural and Multidisciplinary Optimization*, 18:183–192, 1999.
- [85] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, Jamshid Parvizian, M. Ruess, Dominik Schillinger, and Ernst Rank. Fcmlab: A finite cell research toolbox for matlab. *Advances in engineering software*, 2014.
- [86] M. Zhou, R. Fleury, S. Patten, B. Stannard, D. Mylett, and S. Gardner. Topology optimization - practical aspects for industrial applications. In *9th World Congress on Structural and Multidisciplinary Optimization*, 2011.
- [87] M. Zhou and G.I.N. Rozvany. The coc algorithm, part ii: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89:309–336, 1991.
- [88] M. Zhou and G.I.N. Rozvany. On the validity of eso type methods in topology optimization. *Structural and Multidisciplinary Optimization*, 21:80–83, 2001.

-
- [89] S. Zhou and M.Y. Wang. Multimaterial structural topology optimization with a generalized cahn-hilliard model of multiphase transition. *Structural and Multidisciplinary Optimization*, 33:89–111, 2007.
- [90] J.H. Zhu, W.H. Zhang, and L. Xia. Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering*, 2015.

A literature review on topology optimization methods

This appendix will give the reader an elaborate literature review on structural topology optimization. First, the definition and applications of topology optimization will be mentioned. Secondly, the different methods used for topology optimization will be discussed. Thirdly, the developments in large-scale topology optimization will be discussed.

A.1 Applications of structural optimization

The definition of structural optimization as defined by Christensen, and Klarbring [23] states: "Structural optimization is the subject of making an assemblage of materials to sustain loads in the best way". The definition of "best" depends completely on the view-point of the optimization problem. It can mean as light-weight as possible, as stiff as possible, or specify a desired deformation, etc. The optimization problem is most of the times, if not always, bounded by constraints. One can for example look for the stiffest design, for a given mass. Furthermore, it is also possible to have multiple objectives and find a so-called Pareto-optimum [23], which is a weighted sum of the different objective functions.

Structural optimization can be defined in three classes [17][23]:

- **Sizing optimization**

In a sizing problem the goal is to find the optimum of a design variable while the domain of the design is already known. An example of a sizing problem can be finding optimum thickness distribution of a sheet.

- **Shape optimization**

In a shape optimization problem the goal is to find the optimum shape of the domain. So the domain is now the design variable.

- **Topology optimization**

Topology optimization is the most general form of structural optimization. It involves the location and connectivity of the material in the domain.

An example of topology optimization on a continuum structure can be found in Figure A.1 [60], here one sees the load and boundary conditions applied on domain Ω , in the bottom figure one can see the optimized structure for minimum compliance.

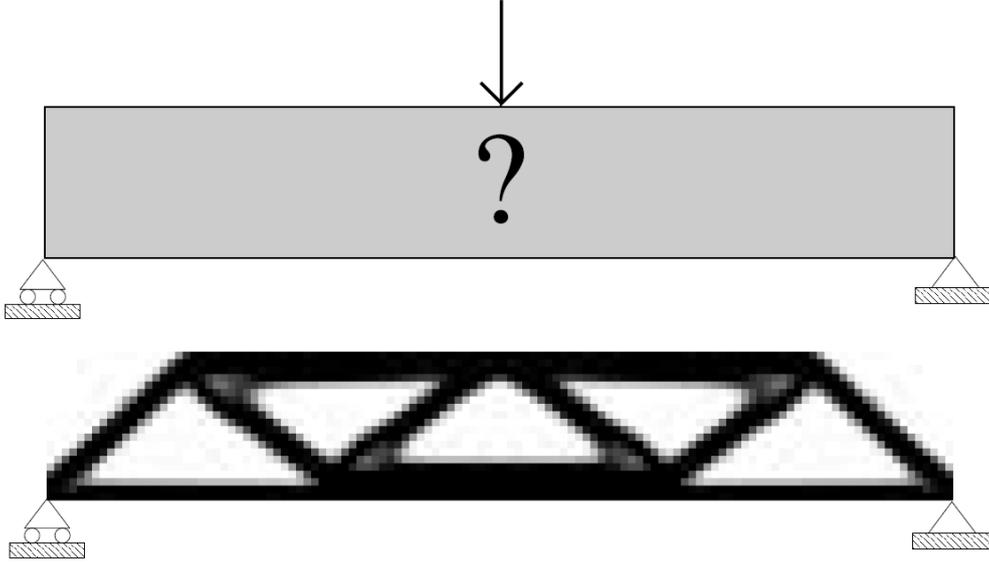


Figure A.1: Example of topology optimization on a MBB beam, the top figure denotes the design domain Ω with the applied loads and boundary conditions, and the bottom figure shows the corresponding optimized structure [60]

Despite the fact that topology optimization is a relatively new technique it already found applications in various industries such as the aerospace and automotive industries [86, 90]. Furthermore, the method has given very promising results for the biomedical industry and the development of Micro ElectroMechanical Systems (MEMS) [17].

A.2 Problem formulation in topology optimization

Since the original paper on topology optimization by Bendsøe and Kikuchi [?], the research field has grown, and it is at the moment one of the most active research area's in structural and multidisciplinary optimization [27]. The development on topology optimization is split up into different directions [64], which will be discussed in more detail later, however the general problem formulation is the same.

Topology optimization can be seen as a material distribution problem. The goal is to find that optimum material distribution $\rho(\mathbf{x})$ that minimizes an objective function \hat{F} . This function is subject to m constraints \hat{G}_i , of which the first is generally a volume constraint. The

mathematical form of the topology optimization problem can be written as [64]:

$$\begin{aligned}
\min_{\rho} : \hat{F}(\rho) &= F(\rho, \mathbf{u}(\rho)) = \int_{\Omega} f(\rho, \mathbf{u}(\rho)) dV \\
\text{s.t.} : \hat{G}_1(\rho) &= \int_{\Omega} \rho(\mathbf{x}) dV - V_{max} \leq 0 \\
&: \hat{G}_i(\rho) = G_i(\rho, \mathbf{u}(\rho)) \leq 0, & i = 2, \dots, m \\
&: \rho(\mathbf{x}) = 0 \text{ or } 1, & \forall \mathbf{x} \in \Omega
\end{aligned} \tag{A.1}$$

where f is a local function, e.g. the strain energy density for minimum compliance problems. The material distribution $\rho(\mathbf{x})$, is either 0 or 1, and \mathbf{u} is solved using the FCM in a separate step. This optimization problem is called a nested topology optimization problem, since the total potential energy is at a stationary point for each optimization step. There exist other approaches where both the displacement and the optimum material distribution are found at the same time, however, this form of optimization will not be considered in this study. It has to be noted that the volume constraint is not necessarily a constraint in all optimization problems, hence the volume can also be an objective that has to be minimized. However, in most academic optimization formulations an optimum topology has to be found, which satisfies this volume constraint.

Various approaches exist to solve the optimization problem specified above, which can be categorized in two main approaches. In the density approach a nodal or element wise density distribution is used to specify the geometry. In the second approach the shape of the design domain is optimized. This shape optimization approach has to allow the possibility of introducing new holes in the material domain to be different from normal shape optimization.

Using the density approach the domain is generally discretized into a large number of elements where the density is described using nodal or element wise design variables. A problem with this discretization is that the density can only be either 0 or 1. This does not allow the use of a gradient based optimization algorithm. This discrete optimization problem can be solved using several approaches as will be discussed in Section A.6. But, when speaking about density approaches, which is the most popular method in topology optimization, it is general to assume that the design variables are continuous. These continuous design variables allow the optimizer to make use of gradient information to solve the problem in an efficient manner.

In the approach where the boundary of the structure is varied the formulation of new holes in the design domain has to be allowed. This can be done with topological derivatives such as the bubble-method introduced by Eschenauer et al [32]. Level set and phase field topology optimization approaches are the most well known methods in which the boundary is varied. However, as is discussed by Sigmund and Maute most of these methods can also be written as a density approach, where the main difference is the way the gradients are calculated [64].

A.3 Topology optimization using a continuous density distribution

In density methods the design domain Ω is predefined and discretized in small elements e . The goal of the optimization is to create the optimum structure Ω_{mat} within Ω . As a design

vector the constitutive tensor \mathbf{C} is chosen, which defines the stiffness per element on the design domain, this means that elements containing solid material get normal stiffness while void elements get zero stiffness. In other words the density based methods decide whether an element should contain material or not. The constitutive matrix is therefore multiplied by a element density ($\rho(\mathbf{x})$), which is allowed to vary between ρ_{min} and ρ_{max} . The optimization algorithm then tries to find the most optimum density distribution for the set optimization problem.

Generally one wants an optimum structure where the material in the density distribution is either void or solid. Therefore the density has to be penalized to steer the solution towards 0-1 solution. Multiple methods exist for the penalization of the density of which the Solid Isotropic Material with Penalization (SIMP) method is the most well known. This method which was independently developed by Bendsøe [14], and Zhou and Rozvany [87] is sometimes referred to as power-law approach as well. In this method intermediate densities are penalized using a penalization factor (q) which makes it unfavorable for the optimizer since only little stiffness is obtained by adding more material [17]. An illustration of the effect of the penalization factor (q) on the density can be seen in Figure A.2. A physical justification for SIMP has been provided by Bendsøe and Sigmund where the penalization factor of the SIMP should fulfill [16],

$$\begin{aligned} q &\geq \max \left\{ \frac{2}{1-\nu_0}, \frac{4}{1+\nu_0} \right\} && \text{(in 2-D),} \\ q &\geq \max \left\{ 15 \frac{1-\nu_0}{7-5\nu_0}, \frac{3}{2} \frac{1-\nu_0}{1-2\nu_0} \right\} && \text{(in 3-D),} \end{aligned} \quad (\text{A.2})$$

Using the penalized density of the SIMP method and substituting it in the stiffness tensor yields,

$$\mathbf{C}(\mathbf{x}) = \rho(\mathbf{x})^q \mathbf{C}_0 \quad (\text{A.3})$$

Where \mathbf{C}_0 denotes the stiffness tensor for the isotropic material.

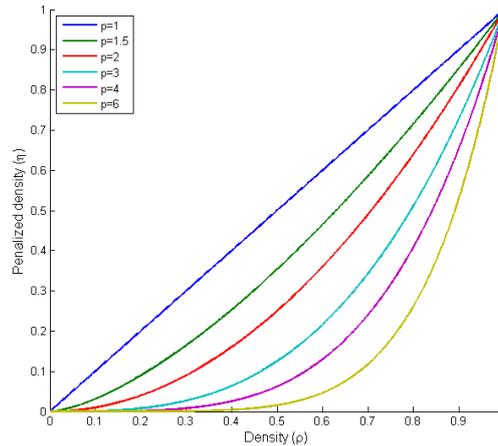


Figure A.2: This figure shows the effect of the different penalization factors q on the density of the material, it can be seen that a high penalty factor makes intermediate densities uneconomical

Using this penalization factor the optimization problem can be solved using several approaches such as, Optimality Criteria (OC), Sequential Linear Programming (SLP), Sequen-

tial Quadratic Programming (SQP), interior point method or the extremely popular Method of Moving Asymptotes (MMA) [69].

Sigmund has introduced a slightly modified version of the SIMP method in which a minimum value for the stiffness is used [61]. The formulation for the element stiffness using the modified SIMP approach then becomes,

$$E_e(x_e) = E_{min} + \rho^q(x_e)(E_0 - E_{min}) \quad (\text{A.4})$$

Where E_0 is the stiffness of the material, and E_{min} is a very small stiffness to avoid singularities in the stiffness matrix. This modified version has a number of advantages over the classical SIMP approach, which include a more straightforward implementation of filters [61, 10].

Apart from SIMP interpolation the Rational Approximation of Material Properties (RAMP) method exists which has been proposed by Stolpe and Svanberg [67], where the main difference that the RAMP method does not have a zero-gradient at $\rho = 0$. Furthermore Bruns has developed the SINH method in which the volume of intermediate density material is penalized [21].

Topology optimization using the density approach requires regularization to control the quality of the solution. The first reason is that the solution is dependent on the mesh. In general the introduction of more holes will increase the efficiency of the structures, therefore a finer mesh will result in a more optimum solution. The second reason is the checkerboard problem, in this case the optimum solution may consist of alternating solid void elements as can be seen in Figure A.3. This is an unrealistic structure and it originates from the fact that the FEM solution overestimates the stiffness of these checkerboards [17, 28].

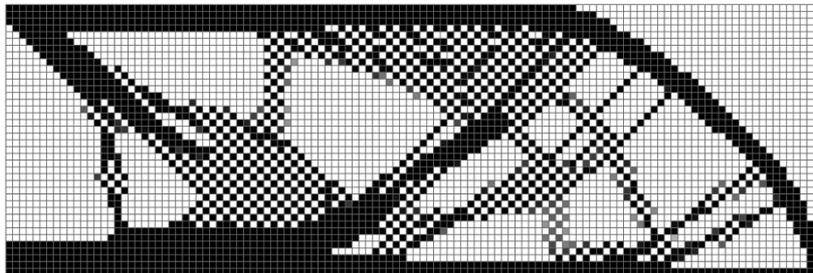


Figure A.3: Solution for a MBB beam to demonstrate the occurrence of the checkerboard problem.

The checkerboard problem can be avoided when higher-order shape functions are used, or when regularization techniques are used to ensure a mesh-independent solution. To ensure mesh independent solutions two things are generally done, either the design variables or sensitivities are filtered, or a constraints can be used to limit the perimeter of the solution. Filter methods remain the most popular methods used for regularization due to their ease of application [27].

Among the first filters that have been used is the sensitivity filter that modifies the element sensitivity values based on the weighted average of the neighboring elements in a specified radius [59]. In a recent paper Sigmund and Maute have proven that the sensitivity filter has a physical basis for compliance minimization problems [63]. Another, very popular filter method is the density filter. In this method the physical density of an element is based on the weighted average of the design variables in neighboring elements in a specified radius [22, 19].

In a recent paper Lazarov and Sigmund developed efficient variations to the density filter using Helmholtz-type differential equations [41].

The consequence of using filters for the density and the sensitivity is formation of 'gray' areas, while in most topology optimization problems a crisp solution is required. To cope with this the density can be projected on a 0-1 solution using different projection schemes, such as the relaxed Heaviside function as proposed by Guest et al. [35], or morphology based techniques as is proposed by Sigmund [61].

A.4 Topology optimization using the level-set method

The level-set method has been developed by Osher and Sethian to model moving boundaries [50], and has later been successfully implemented for topology optimization [51, 4, 79]. In the level-set method the boundaries are represented using the level-set function $\phi(\mathbf{x})$. The boundary can be seen as the zero-level of the - set function,

$$\rho(\mathbf{x}) = \begin{cases} 0 & \forall \mathbf{x} \in \Omega \quad \phi(\mathbf{x}) < 0 \\ 1 & \forall \mathbf{x} \in \Omega \quad \phi(\mathbf{x}) \geq 0 \end{cases} \quad (\text{A.5})$$

In most cases the level-set function ($\phi(\mathbf{x})$) is updated using the Hamilton-Jacobi equation, which is normally augmented with diffusive and reactive terms to regularize the optimization problem and to nucleate new holes in the design [64]. Using the level-set method a smooth boundary representation can be achieved, however, most level-set method rely on finite elements therefore the boundaries must be discretized. In a recent paper by van Dijk et al. three approaches are discussed to map the geometry on the finite element mesh, which can be seen in FigureA.4 [74].

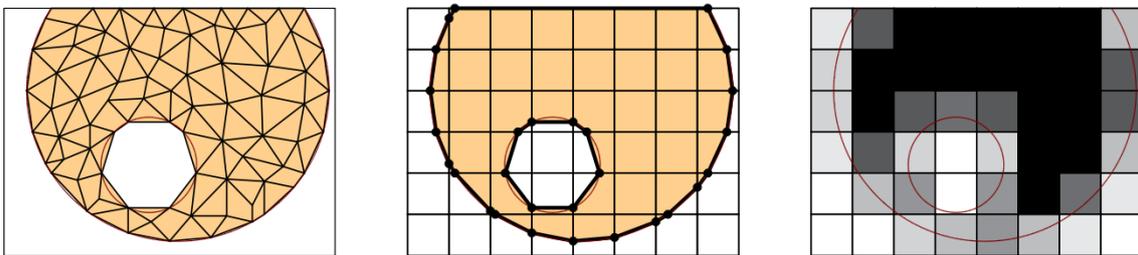


Figure A.4: "Examples of different types of geometry mapping, (a) Conforming: Only the material domain is discretized. (b) Immersed Boundary Technique: Boundary conditions are enforced locally at the interface. (c) Density-based: In each finite element an element density indicates the amount of material" [74]

Using the density mapping the material is described by an intermediate density field, just as in continuous density approaches. It can furthermore be noticed that the FCM is actually an immersed boundary method, and that mapping the material on a design domain, would be straightforward using the FCM.

A.5 Topology optimization using the phase field approach

The phase field method has been proposed for structural optimization by Bourdin and Chambolle [20]. In this method the material in domain Ω consist of two phases, called A and B ,

which are specified using the phase field function ϕ [27]. Phase A correspond to zero density and phase B to the density of the material considered. The boundary between the phases is a continuously varying region of finite thickness ξ as can be seen in Figure A.5 [73].

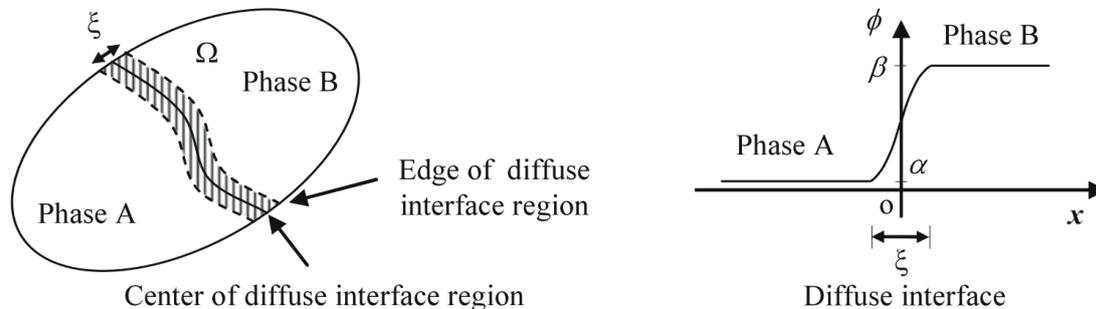


Figure A.5: "Examples of the phase field function: (a) A 2D Example represented by the phase field function, (b) A 1D illustration of the phase field function" [73]

In topology optimization using the phase field method the interface region is modified using the phase field equation. The objective that is minimized consists of a double well function that penalizes the intermediate density values. Minimization of the function is obtained using the Cahn-Hilliard equation as is proposed by Whang and Zhou [89]. As is discussed by Sigmund and Maute phase field approaches are known for slow convergence rates [64].

A.6 Topology optimization using discrete evolutionary approaches

Of the discrete approaches the most well known discrete approaches are the Evolutionary Structural Optimization (ESO) and the Bi-directional ESO (BESO). The ESO approach which has been proposed by Xie and Steven is a hard-kill method for topology optimization [81, 82]. In this method material is gradually removed until an optimum solution is obtained. In the BESO method proposed by Querrin et al. material can be added as well [55, 84]. Although initially the method has been fully heuristic nowadays the methods use an adjoint gradient analysis and filtering similar to density methods [11, 12, 36]. In a recent review paper Sigmund and Maute discuss that these BESO methods should not be treated as separate methods but rather as a discrete updated version of the standard SIMP scheme [64].

ESO and BESO methods have received a lot of critiques. In a paper by Rozvany and Zhou it is shown that ESO may result in a highly non-optimal design [88]. In a more recent paper Rozvany gives an overview of all the critiques on the ESO methods and concludes that the method can be seen as fully heuristic without any verification [56]. Furthermore, a major problem with ESO/BESO methods is the fact that it cannot take multiple non-linear constraints into account.

A.7 Topology optimization for large scale optimization problems

Large scale topology optimization comes with a high computational cost. Therefore it is very important to perform this optimization as efficient as possible. The computational effort required for this optimization is driven by three different aspects: the complexity of the model, the FE-analysis and the optimization process [76].

In the past years a lot of research has been performed to decrease the computational effort of these three aspects. The goal of this section is to give an overview of the current state of research in large-scale topology optimization. Furthermore the use of parallel topology optimization will be discussed since this technology can greatly reduce the solution time [2].

A.7.1 Reduction of model complexity

In terms of model complexity the amount of degrees of freedom in the model is most of the times an indication of the computational cost involved in the analysis process since it influences the size of the stiffness matrix. However the topology of the structure has an important effect on the size of the stiffness matrix as well. A complex 3D structure usually requires a larger bandwidth than a slender beam [76].

To reduce the complexity of the model Kim and Yoon suggested a multi-resolution multi-scale topology optimization method using wavelet transforms [39]. In this method the optimization is started at a low resolution, requiring less time to optimize. Afterwards this solution is used as a starting point for optimization using a higher resolution. This process is repeated till the desired resolution is reached. Instead of a design vector using the element densities, wavelet-based design variables are used, which has several advantages in multi-resolution multi-scale topology optimization, such as preventing the occurrence of checkerboard patterns and geometry control as is discussed by Poulsen [54].

Instead of increasing the resolution of the whole design domain Stainko has proposed a method where the mesh is adaptively refined along the interface between solid and void [66]. This method results in a fine-resolution design at a relatively low computational expense. Kim and Kwak have developed a method in which the optimum design space has to be found [37]. This means that not only an optimum structure is found in a design space, but that the design space evolves as well. The number of design variables is unknown in the beginning and has to be obtained as well.

Another method to reduce the complexity of the model yet retaining a high resolution for the solution is Multi-resolution Topology Optimization (MTOP) described by Nguyen et al. [47]. In this method the density resolution is defined on a finer mesh than the displacements as can be seen in Figure A.6.

The effect of this finer density mesh on the analysis mesh is included in the integration process. In a more recent paper an Adaptive Multi-resolution Topology Optimization (AMTOP) method is introduced in which the design variable and density fields can be adjusted and refined [48]. It has to be noted that the MTOP can be implemented very simply using the FCM since this method already consists of a separate geometry and analysis description.

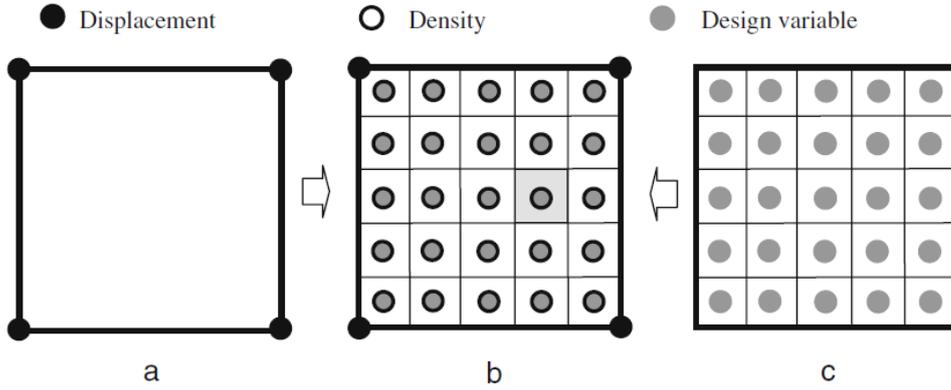


Figure A.6: Decoupling of the displacement mesh, and design variable mesh using MTOP. (a) Displacement mesh, (b) Superposed meshes, (c) Design Variable mesh [47]

A.7.2 Reduction of the FE-analysis time

The solution of the FE-equation requires the most computational effort in nested topology optimization [7], therefore numerous efforts have been made to reduce the analysis time. Wang et al. make use of the fact that the linear systems in the optimization procedure change slowly during every step [80]. Therefore the solution time can be reduced by recycling parts of the previous linear systems. Since an iterative solver is used it has the additional advantage that it can be easily parallelized which is a great benefit for larger problems.

Amir et al. reformulate the optimization problem using an approximation of the displacement vector [7]. It is shown that even relatively rough approximations are acceptable, since the errors are taken into account in the sensitivity analysis. In a later paper this approach has been combined with a Krylov subspace solver reducing the computation time even further [9]. In a more recent paper Amir and Sigmund discussed an approach which required only one matrix factorization during the whole design process [8]. This method uses a small number of iterative corrections for each design cycle resulting in a reduction of computational cost by one order of magnitude without affecting the outcome of the optimization process.

Furthermore, Amir et al. have shown that the use of a Multi-Grid Conjugate Gradients method (MGCG) used for solving the nested analysis equations can greatly reduce the computation time for topology optimization problem [6]. Furthermore they showed that a MATLAB implementation of this method can solve a 3D topology optimization with 400.000 degrees of freedom on a standard PC in less than 15 minutes.

A.7.3 Increase in optimization efficiency

An increase in optimization efficiency can be obtained using several methods, first of all the design variables can be reformulated to make the problem more efficient or the problem can be formulated in a different way. Guest et al. decouple the finite element mesh and the design variable mesh [34], where a Heaviside projection filter is used to project the design variables onto multiple elements. This means that a single design variable influences several elements. The computational savings using this method are small for minimum compliance problems with a low number of constraints. However, when many local constraints are used this method substantially reduces the computational effort.

Kim et al. have proposed the Reducible Design Variable Method (RDVM) which gradually reduces the number of design variables over iterations [70]. In a fixed domain where the number of design variables equal the number of elements the design variables are 'fixed' for element which have converged to a density value. This process is repeated until the number of design variables is equal to zero. Suresh very recently proposed a method to generate sets of pareto-optimal topologies in a single optimization process [68], making use of Matrix-free Krylov iterations. The main advantage of this method is that the set of pareto-optimal topologies allows the engineer to see the solution for multiple volume fractions using one optimization process.

A.7.4 Parallel topology optimization

When applying large scale topology optimization the required computational effort is very large. As is mentioned in [2]: "The only feasible way to obtain a solution within a reasonable amount of time is to use parallel computations in order to speed up the solution process". Borvall and Petersson are the first to describe topology optimization using parallel computing techniques [18]. Other contributions to parallel topology optimization can be found in [38, 75, 43, 3, 33]. Furthermore, Wadbro and Berggren demonstrate the potential of GPU's for parallel topology optimization in [77].

When applying parallel computation one should consider Amdahl's law [5]. Which states that for a fixed problem the increase in speed is limited by the sequential part of the code. This means that the code fully has to be parallelized to increase a linear increase in speed with the number of CPU's. A depiction of Amdahl's law can be seen in Figure A.7 [2]. It can

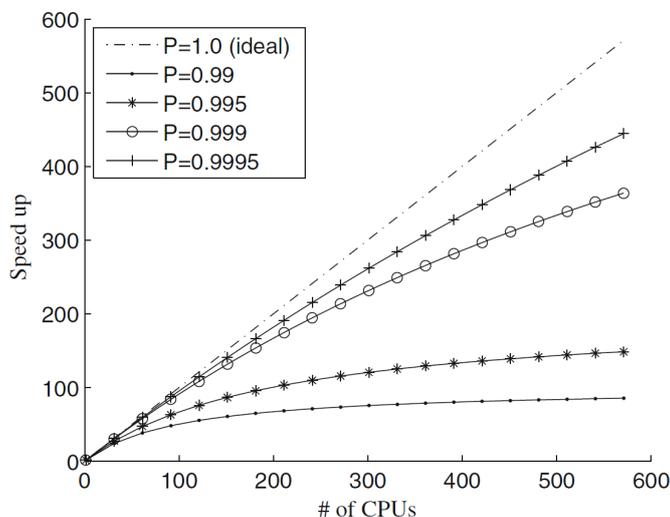


Figure A.7: "Amdahl's law for parallel speedup" [2]

be seen that even when a small part of the code that is not parallelized a large limit on the increase in speed is posed when a large number of CPU's are used. This means that not only the FE-analysis equations have to be parallelized but the optimizer as well. In one of the most recent papers on parallel topology optimization AAGE and LAZAROV have parallelized the method of moving asymptotes [2].

Appendix B

Minimum compliance experiments

On the next pages, the results of the optimization examples using the MBB-beam are shown. Two different discretization are used (280 by 60 and 240 by 80), and five different types of filter methods are used. The corresponding results are shown in Table B.1 - B.10.

Table B.1: Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	22082	408	0.40	1.13E+06	1.18E+06	12.65
1	2	5642	422	0.40	1.12E+06	1.18E+06	12.71
1	3	2562	287	0.40	1.09E+06	1.71E+06	14.30
1	4	1472	289	0.40	1.08E+06	2.32E+07	18.57
1	5	962	352	0.40	1.05E+06	3.73E+07	21.46
1	6	682	194	0.40	1.03E+06	1.65E+08	23.05
1	10	266	136	0.40	9.18E+05	1.74E+11	14.94
2	1	87362	412	0.40	1.17E+06	1.19E+06	12.49
2	2	22082	210	0.40	1.15E+06	1.19E+06	12.49
2	3	9922	392	0.40	1.13E+06	1.18E+06	12.86
2	4	5642	649	0.40	1.12E+06	1.18E+06	12.86
2	5	3650	286	0.40	1.11E+06	1.20E+06	12.59
2	6	2562	339	0.40	1.10E+06	2.03E+06	15.14
2	10	962	188	0.40	1.02E+06	3.86E+07	17.30
3	2	49322	509	0.40	1.16E+06	1.18E+06	12.89
3	3	22082	325	0.40	1.15E+06	1.18E+06	12.76
3	4	12512	378	0.40	1.14E+06	1.18E+06	12.73
3	5	8066	441	0.40	1.13E+06	1.18E+06	12.72
3	6	5642	349	0.40	1.13E+06	1.19E+06	12.62
3	10	2090	489	0.40	1.10E+06	1.76E+06	13.98
4	2	87362	437	0.40	1.17E+06	1.19E+06	12.44
4	3	39042	383	0.40	1.16E+06	1.18E+06	12.73
4	4	22082	365	0.40	1.15E+06	1.18E+06	12.66
4	5	14210	434	0.40	1.14E+06	1.18E+06	12.81
4	6	9922	388	0.40	1.14E+06	1.18E+06	12.70
4	10	3650	241	0.40	1.13E+06	1.23E+06	13.24
5	2	136202	330	0.40	1.19E+06	1.19E+06	12.29
5	3	60802	410	0.40	1.17E+06	1.18E+06	12.85
5	4	34352	285	0.40	1.16E+06	1.19E+06	12.63
5	5	22082	439	0.40	1.15E+06	1.18E+06	12.77
5	6	15402	273	0.40	1.15E+06	1.19E+06	12.54
5	10	5642	366	0.40	1.13E+06	1.19E+06	12.72

Table B.2: Results of the stability experiments on the MBB-beam example. A discretization of 240 by 80 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	39042	601	4.00E-01	1.11E+06	1.16E+06	10.97
1	2	9922	637	4.00E-01	1.10E+06	1.16E+06	11.00
1	4	2562	455	4.00E-01	1.06E+06	8.65E+06	13.06
1	5	1666	559	4.00E-01	1.04E+06	3.16E+07	17.24
1	8	682	291	4.00E-01	9.89E+05	3.08E+08	18.93
1	10	450	245	4.00E-01	9.45E+05	8.06E+10	15.67
2	1	154882	562	4.00E-01	1.15E+06	1.16E+06	11.01
2	2	39042	556	4.00E-01	1.12E+06	1.16E+06	11.10
2	4	9922	647	4.00E-01	1.10E+06	1.16E+06	10.91
2	5	6402	458	4.00E-01	1.10E+06	1.17E+06	10.63
2	8	2562	288	4.00E-01	1.07E+06	3.68E+07	19.41
2	10	1666	548	4.00E-01	1.03E+06	3.88E+07	16.35
3	2	87362	528	4.00E-01	1.14E+06	1.16E+06	10.95
3	4	22082	552	4.00E-01	1.12E+06	1.16E+06	10.93
3	5	14210	498	4.00E-01	1.12E+06	1.16E+06	11.01
3	8	5642	649	4.00E-01	1.10E+06	1.16E+06	10.77
3	10	3650	368	4.00E-01	1.09E+06	2.28E+06	12.46
4	2	154882	571	4.00E-01	1.15E+06	1.16E+06	10.95
4	4	39042	487	4.00E-01	1.13E+06	1.16E+06	10.88
4	5	25090	484	4.00E-01	1.12E+06	1.16E+06	10.91
4	8	9922	482	4.00E-01	1.11E+06	1.16E+06	10.99
4	10	6402	501	4.00E-01	1.11E+06	1.17E+06	11.07
5	2	241602	591	4.00E-01	1.15E+06	1.16E+06	10.86
5	4	60802	531	4.00E-01	1.14E+06	1.16E+06	10.87
5	5	39042	544	4.00E-01	1.13E+06	1.16E+06	10.87
5	8	15402	512	4.00E-01	1.12E+06	1.16E+06	11.03
5	10	9922	488	4.00E-01	1.12E+06	1.16E+06	11.07

Table B.3: Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	22082	712	4.00E-01	1.17E+06	1.22E+06	15.49
1	2	5642	584	4.00E-01	1.16E+06	1.22E+06	15.47
1	3	2562	874	4.00E-01	1.16E+06	1.58E+06	17.74
1	4	1472	657	4.00E-01	1.16E+06	4.68E+07	25.55
1	5	962	973	4.00E-01	1.15E+06	5.48E+07	28.68
1	6	682	310	4.00E-01	1.14E+06	6.49E+07	31.85
1	10	266	243	4.00E-01	9.69E+05	1.41E+11	19.97
2	1	87362	1048	4.00E-01	1.19E+06	1.21E+06	13.15
2	2	22082	714	4.00E-01	1.18E+06	1.22E+06	15.39
2	3	9922	991	4.00E-01	1.17E+06	1.22E+06	15.44
2	4	5642	659	4.00E-01	1.17E+06	1.22E+06	15.52
2	5	3650	691	4.00E-01	1.15E+06	1.21E+06	15.12
2	6	2562	4222	4.00E-01	1.18E+06	3.68E+06	20.62
2	10	962	735	4.00E-01	1.11E+06	5.27E+07	25.18
3	2	49322	866	4.00E-01	1.19E+06	1.21E+06	13.86
3	3	22082	806	4.00E-01	1.19E+06	1.22E+06	15.45
3	4	12512	694	4.00E-01	1.18E+06	1.22E+06	15.44
3	5	8066	989	4.00E-01	1.17E+06	1.22E+06	14.96
3	6	5642	1440	4.00E-01	1.16E+06	1.21E+06	13.48
3	10	2090	1002	4.00E-01	1.18E+06	3.75E+07	22.19
4	2	87362	1319	4.00E-01	1.20E+06	1.21E+06	13.82
4	3	39042	1233	4.00E-01	1.19E+06	1.21E+06	13.82
4	4	22082	994	4.00E-01	1.19E+06	1.22E+06	15.41
4	5	14210	1077	4.00E-01	1.19E+06	1.22E+06	15.51
4	6	9922	968	4.00E-01	1.18E+06	1.22E+06	15.43
4	10	3650	641	4.00E-01	1.17E+06	1.26E+06	16.18
5	2	136202	960	4.00E-01	1.21E+06	1.21E+06	14.08
5	3	60802	914	4.00E-01	1.20E+06	1.21E+06	13.84
5	4	34352	1132	4.00E-01	1.19E+06	1.21E+06	13.88
5	5	22082	966	4.00E-01	1.19E+06	1.21E+06	13.87
5	6	15402	850	4.00E-01	1.18E+06	1.21E+06	13.97
5	10	5642	796	4.00E-01	1.17E+06	1.21E+06	14.97

Table B.4: Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	39042	1000	4.00E-01	1.14E+06	1.18E+06	12.04
1	2	9922	1000	4.00E-01	1.13E+06	1.18E+06	12.02
1	4	2562	942	4.00E-01	1.15E+06	5.23E+07	23.64
1	5	1666	719	4.00E-01	1.15E+06	5.62E+07	27.63
1	8	682	448	4.00E-01	1.06E+06	7.43E+07	24.94
1	10	450	304	4.00E-01	1.01E+06	7.54E+08	22.19
2	1	154882	945	4.00E-01	1.17E+06	1.18E+06	12.01
2	2	39042	1000	4.00E-01	1.15E+06	1.18E+06	11.99
2	4	9922	1000	4.00E-01	1.13E+06	1.18E+06	12.28
2	5	6402	1000	4.00E-01	1.14E+06	1.20E+06	12.39
2	8	2562	1000	4.00E-01	1.16E+06	6.41E+07	25.24
2	10	1666	624	4.00E-01	1.11E+06	5.24E+07	22.66
3	2	87362	1000	4.00E-01	1.16E+06	1.18E+06	12.12
3	4	22082	1000	4.00E-01	1.15E+06	1.18E+06	11.96
3	5	14210	1000	4.00E-01	1.15E+06	1.18E+06	12.10
3	8	5642	830	4.00E-01	1.14E+06	1.23E+06	12.53
3	10	3650	1000	4.00E-01	1.17E+06	5.06E+06	20.05
4	2	154882	996	4.00E-01	1.17E+06	1.18E+06	12.17
4	4	39042	991	4.00E-01	1.16E+06	1.18E+06	12.03
4	5	25090	1000	4.00E-01	1.15E+06	1.18E+06	12.00
4	8	9922	1000	4.00E-01	1.15E+06	1.19E+06	11.92
4	10	6402	932	4.00E-01	1.14E+06	1.20E+06	12.42
5	2	241602	1000	4.00E-01	1.18E+06	1.18E+06	12.04
5	4	60802	1000	4.00E-01	1.17E+06	1.18E+06	12.03
5	5	39042	1000	4.00E-01	1.16E+06	1.18E+06	12.14
5	8	15402	936	4.00E-01	1.15E+06	1.19E+06	11.93
5	10	9922	1000	4.00E-01	1.15E+06	1.19E+06	12.38

Table B.5: Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	22082	383	4.00E-01	1.12E+06	1.18E+06	2.27
1	2	5642	406	4.00E-01	1.05E+06	1.16E+06	0.96
1	3	2562	374	4.00E-01	1.02E+06	4.77E+07	1.45
1	4	1472	403	4.00E-01	9.69E+05	4.32E+07	1.27
1	5	962	396	4.00E-01	9.35E+05	3.91E+07	1.47
1	6	682	367	4.00E-01	9.18E+05	5.39E+07	1.38
1	10	266	345	4.00E-01	8.39E+05	2.77E+10	0.95
2	1	87362	414	4.00E-01	1.11E+06	1.12E+06	1.95
2	2	22082	507	4.00E-01	1.14E+06	1.19E+06	1.75
2	3	9922	386	4.00E-01	1.06E+06	1.44E+06	1.45
2	4	5642	380	4.00E-01	1.05E+06	1.20E+06	1.25
2	5	3650	396	4.00E-01	1.03E+06	1.87E+07	1.27
2	6	2562	379	4.00E-01	1.01E+06	4.45E+07	1.37
2	10	962	330	4.00E-01	9.39E+05	4.10E+07	0.88
3	2	49322	367	4.00E-01	1.10E+06	1.12E+06	1.60
3	3	22082	528	4.00E-01	1.11E+06	1.16E+06	1.80
3	4	12512	377	4.00E-01	1.07E+06	1.14E+06	1.67
3	5	8066	443	4.00E-01	1.11E+06	1.20E+06	2.78
3	6	5642	383	4.00E-01	1.05E+06	1.48E+06	1.28
3	10	2090	384	4.00E-01	1.01E+06	4.93E+07	1.45
4	2	87362	517	4.00E-01	1.16E+06	1.17E+06	1.78
4	3	39042	493	4.00E-01	1.14E+06	1.18E+06	2.09
4	4	22082	443	4.00E-01	1.13E+06	1.18E+06	2.06
4	5	14210	380	4.00E-01	1.09E+06	1.15E+06	1.81
4	6	9922	582	4.00E-01	1.09E+06	1.16E+06	1.61
4	10	3650	422	4.00E-01	1.05E+06	3.32E+06	1.09
5	2	136202	538	4.00E-01	1.14E+06	1.15E+06	1.80
5	3	60802	503	4.00E-01	1.13E+06	1.15E+06	1.94
5	4	34352	496	4.00E-01	1.13E+06	1.16E+06	1.91
5	5	22082	471	4.00E-01	1.13E+06	1.17E+06	1.87
5	6	15402	473	4.00E-01	1.11E+06	1.17E+06	1.72
5	10	5642	369	4.00E-01	1.06E+06	1.44E+06	1.25

Table B.6: Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	39042	415	4.00E-01	1.06E+06	1.11E+06	2.14
1	2	9922	389	4.00E-01	1.04E+06	1.13E+06	1.40
1	4	2562	363	4.00E-01	9.75E+05	4.62E+07	2.07
1	5	1666	421	4.00E-01	9.43E+05	4.94E+07	2.10
1	8	682	348	4.00E-01	8.81E+05	5.14E+07	1.59
1	10	450	364	4.00E-01	8.61E+05	4.87E+09	1.26
2	1	154882	406	4.00E-01	1.10E+06	1.11E+06	2.01
2	2	39042	356	4.00E-01	1.07E+06	1.11E+06	1.81
2	4	9922	342	4.00E-01	1.05E+06	1.13E+06	1.49
2	5	6402	401	4.00E-01	1.04E+06	7.89E+06	1.54
2	8	2562	377	4.00E-01	9.62E+05	4.96E+07	1.69
2	10	1666	368	4.00E-01	9.45E+05	4.34E+07	1.38
3	2	87362	405	4.00E-01	1.09E+06	1.11E+06	1.96
3	4	22082	348	4.00E-01	1.07E+06	1.12E+06	2.11
3	5	14210	409	4.00E-01	1.06E+06	1.12E+06	1.44
3	8	5642	372	4.00E-01	1.04E+06	1.60E+07	1.66
3	10	3650	386	4.00E-01	1.01E+06	5.05E+07	2.32
4	2	154882	393	4.00E-01	1.10E+06	1.11E+06	1.95
4	4	39042	359	4.00E-01	1.08E+06	1.11E+06	2.12
4	5	25090	363	4.00E-01	1.07E+06	1.11E+06	1.75
4	8	9922	423	4.00E-01	1.05E+06	1.11E+06	1.42
4	10	6402	397	4.00E-01	1.05E+06	1.85E+06	1.62
5	2	241602	363	4.00E-01	1.11E+06	1.11E+06	2.15
5	4	60802	375	4.00E-01	1.09E+06	1.11E+06	2.24
5	5	39042	413	4.00E-01	1.08E+06	1.11E+06	1.76
5	8	15402	375	4.00E-01	1.07E+06	1.13E+06	2.34
5	10	9922	378	4.00E-01	1.05E+06	1.11E+06	1.38

Table B.7: Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	22082	284	3.97E-01	1.10E+06	1.15E+06	0.94
1	2	5642	315	3.98E-01	1.07E+06	1.19E+06	0.93
1	3	2562	307	3.97E-01	1.05E+06	3.05E+06	0.92
1	4	1472	307	3.98E-01	9.92E+05	3.62E+07	0.97
1	5	962	257	3.91E-01	9.47E+05	2.91E+07	0.98
1	6	682	303	3.98E-01	9.04E+05	3.05E+07	0.93
1	10	266	303	3.99E-01	8.49E+05	8.60E+11	0.75
2	1	87362	279	3.98E-01	1.13E+06	1.15E+06	0.96
2	2	22082	304	3.98E-01	1.11E+06	1.15E+06	0.92
2	3	9922	304	3.98E-01	1.10E+06	1.16E+06	0.88
2	4	5642	307	3.97E-01	1.09E+06	1.18E+06	0.92
2	5	3650	308	3.97E-01	1.06E+06	1.48E+06	0.89
2	6	2562	311	3.98E-01	1.03E+06	4.05E+07	0.97
2	10	962	303	3.98E-01	9.48E+05	3.07E+07	0.88
3	2	49322	284	4.00E-01	1.12E+06	1.14E+06	1.00
3	3	22082	307	4.00E-01	1.11E+06	1.15E+06	0.94
3	4	12512	310	3.99E-01	1.10E+06	1.15E+06	0.94
3	5	8066	306	3.98E-01	1.10E+06	1.16E+06	0.90
3	6	5642	307	3.98E-01	1.09E+06	1.17E+06	0.88
3	10	2090	305	3.98E-01	1.01E+06	3.93E+07	0.98
4	2	87362	288	4.00E-01	1.13E+06	1.14E+06	1.11
4	3	39042	331	4.00E-01	1.12E+06	1.14E+06	1.22
4	4	22082	294	3.99E-01	1.11E+06	1.15E+06	0.97
4	5	14210	305	3.99E-01	1.11E+06	1.15E+06	0.95
4	6	9922	321	3.98E-01	1.10E+06	1.15E+06	0.94
4	10	3650	317	3.97E-01	1.07E+06	1.82E+06	1.00
5	2	136202	284	3.99E-01	1.14E+06	1.14E+06	1.00
5	3	60802	292	4.00E-01	1.13E+06	1.14E+06	1.14
5	4	34352	330	4.00E-01	1.12E+06	1.14E+06	1.25
5	5	22082	314	3.99E-01	1.12E+06	1.15E+06	1.00
5	6	15402	382	4.00E-01	1.11E+06	1.15E+06	1.09
5	10	5642	308	3.97E-01	1.09E+06	1.22E+06	0.98

Table B.8: Results of the stability experiments on the MBB-beam example. A discretization of 240 by 120 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	39042	287	3.98E-01	1.11E+06	1.15E+06	1.42
1	2	9922	304	3.98E-01	1.09E+06	1.16E+06	1.31
1	4	2562	308	3.98E-01	1.01E+06	3.71E+07	1.58
1	5	1666	316	3.97E-01	9.69E+05	2.84E+07	1.47
1	8	682	303	3.97E-01	9.04E+05	2.70E+07	1.40
1	10	450	317	3.91E-01	8.91E+05	7.44E+11	1.28
2	1	154882	295	3.99E-01	1.13E+06	1.15E+06	1.46
2	2	39042	304	3.99E-01	1.12E+06	1.15E+06	1.46
2	4	9922	304	3.98E-01	1.09E+06	1.16E+06	1.41
2	5	6402	320	3.98E-01	1.09E+06	1.50E+06	1.42
2	8	2562	314	3.99E-01	9.84E+05	3.09E+07	1.49
2	10	1666	304	3.97E-01	9.67E+05	2.43E+07	1.44
3	2	87362	294	4.00E-01	1.13E+06	1.15E+06	1.71
3	4	22082	308	3.99E-01	1.11E+06	1.15E+06	1.48
3	5	14210	330	3.99E-01	1.10E+06	1.15E+06	1.36
3	8	5642	317	3.98E-01	1.08E+06	1.61E+06	1.44
3	10	3650	308	3.98E-01	1.05E+06	3.04E+07	1.54
4	2	154882	326	4.00E-01	1.14E+06	1.15E+06	2.01
4	4	39042	305	3.99E-01	1.12E+06	1.15E+06	1.46
4	5	25090	311	3.99E-01	1.11E+06	1.15E+06	1.49
4	8	9922	318	3.98E-01	1.10E+06	1.20E+06	1.43
4	10	6402	327	3.98E-01	1.08E+06	1.65E+06	1.43
5	2	241602	302	4.00E-01	1.15E+06	1.15E+06	1.81
5	4	60802	306	4.00E-01	1.13E+06	1.15E+06	1.48
5	5	39042	346	4.00E-01	1.12E+06	1.15E+06	1.56
5	8	15402	370	4.00E-01	1.11E+06	1.15E+06	1.45
5	10	9922	321	3.98E-01	1.10E+06	1.18E+06	1.37

Table B.9: Results of the stability experiments on the MBB-beam example. A discretization of 180 by 60 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	22082	328	4.01E-01	1.16E+06	1.21E+06	0.85
1	2	5642	316	3.94E-01	1.16E+06	1.23E+06	1.32
1	3	2562	312	4.02E-01	1.12E+06	1.39E+06	0.99
1	4	1472	350	3.97E-01	1.11E+06	1.76E+06	1.76
1	5	962	312	4.01E-01	1.03E+06	5.90E+07	1.30
1	6	682	395	4.00E-01	9.69E+05	5.52E+07	2.59
1	10	266	418	4.08E-01	8.43E+05	3.53E+12	1.03
2	1	87362	310	4.01E-01	1.19E+06	1.20E+06	0.77
2	2	22082	319	4.00E-01	1.18E+06	1.21E+06	1.17
2	3	9922	315	4.00E-01	1.16E+06	1.20E+06	1.16
2	4	5642	315	3.99E-01	1.15E+06	1.21E+06	1.03
2	5	3650	327	4.00E-01	1.15E+06	1.33E+06	0.98
2	6	2562	309	3.99E-01	1.16E+06	1.71E+06	1.26
2	10	962	491	3.87E-01	1.06E+06	6.09E+07	5.54
3	2	49322	309	4.01E-01	1.19E+06	1.21E+06	0.85
3	3	22082	303	4.00E-01	1.18E+06	1.21E+06	0.92
3	4	12512	306	4.01E-01	1.17E+06	1.20E+06	0.79
3	5	8066	316	4.00E-01	1.17E+06	1.22E+06	1.02
3	6	5642	316	4.02E-01	1.14E+06	1.19E+06	0.86
3	10	2090	356	3.97E-01	1.17E+06	1.57E+06	1.66
4	2	87362	312	4.01E-01	1.20E+06	1.21E+06	0.73
4	3	39042	307	4.01E-01	1.19E+06	1.21E+06	0.75
4	4	22082	306	4.02E-01	1.17E+06	1.20E+06	0.79
4	5	14210	309	4.00E-01	1.18E+06	1.21E+06	0.98
4	6	9922	309	3.99E-01	1.17E+06	1.21E+06	0.99
4	10	3650	336	4.05E-01	1.16E+06	1.21E+06	1.10
5	2	136202	329	4.00E-01	1.21E+06	1.21E+06	0.83
5	3	60802	306	4.00E-01	1.20E+06	1.22E+06	0.75
5	4	34352	329	4.01E-01	1.19E+06	1.20E+06	0.70
5	5	22082	314	4.01E-01	1.18E+06	1.21E+06	0.77
5	6	15402	343	4.00E-01	1.17E+06	1.20E+06	0.74
5	10	5642	339	4.05E-01	1.16E+06	1.20E+06	0.91

Table B.10: Results of the stability experiments on the MBB-beam example. A discretization of 240 by 80 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-] [-]	C [J]	M_{nd} [%]
1	1	39042	311	4.02E-01	1.18E+06	1.22E+06	0.73
1	2	9922	308	4.01E-01	1.19E+06	1.25E+06	0.98
1	4	2562	369	4.04E-01	1.11E+06	1.60E+06	1.35
1	5	1666	319	4.05E-01	1.07E+06	5.68E+07	1.97
1	8	682	315	4.06E-01	9.18E+05	4.13E+11	2.35
1	10	450	311	4.02E-01	8.85E+05	7.38E+11	1.87
2	1	154882	305	4.01E-01	1.23E+06	1.24E+06	0.80
2	2	39042	317	4.01E-01	1.19E+06	1.23E+06	0.97
2	4	9922	348	4.00E-01	1.20E+06	1.25E+06	1.13
2	5	6402	389	4.03E-01	1.16E+06	1.23E+06	0.86
2	8	2562	459	4.07E-01	1.09E+06	5.35E+07	1.33
2	10	1666	365	3.93E-01	1.07E+06	6.20E+07	5.41
3	2	87362	346	4.01E-01	1.22E+06	1.23E+06	1.05
3	4	22082	342	4.00E-01	1.21E+06	1.25E+06	1.13
3	5	14210	350	4.02E-01	1.20E+06	1.24E+06	0.93
3	8	5642	330	4.01E-01	1.17E+06	1.23E+06	1.24
3	10	3650	386	3.96E-01	1.19E+06	1.65E+06	1.90
4	2	154882	339	4.00E-01	1.25E+06	1.25E+06	1.00
4	4	39042	317	4.02E-01	1.20E+06	1.23E+06	0.63
4	5	25090	310	4.02E-01	1.19E+06	1.22E+06	0.88
4	8	9922	335	4.00E-01	1.22E+06	1.26E+06	1.15
4	10	6402	348	4.01E-01	1.18E+06	1.41E+06	1.08
5	2	241602	313	4.02E-01	1.23E+06	1.23E+06	0.90
5	4	60802	342	4.00E-01	1.24E+06	1.25E+06	0.83
5	5	39042	395	4.03E-01	1.21E+06	1.23E+06	0.93
5	8	15402	351	3.98E-01	1.23E+06	1.26E+06	1.32
5	10	9922	376	3.97E-01	1.22E+06	1.26E+06	1.31

Appendix C

Minimum displacement experiments

On the next pages, the results of the optimization examples using the compliant force inverter example are shown. Two different discretization are used (280 by 60 and 240 by 80), and five different types of filter methods are used. The corresponding results are shown in Table C.1 - C.10.

Table C.1: Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	14762	1000	3.00E-01	-2.49E-02	-2.16E-02	12.67
1	2	3782	1000	2.98E-01	-9.90E-03	-1.14E-02	11.83
1	3	1722	1000	2.95E-01	-2.13E-02	-1.04E-02	15.96
1	4	992	1000	2.99E-01	-1.81E-02	-1.54E-02	12.11
1	5	650	1000	2.85E-01	6.17E-09	2.03E-09	19.64
1	6	462	1000	2.93E-01	-2.12E-03	1.43E-07	12.81
1	10	182	507	3.00E-01	-6.02E-03	2.46E-12	9.57
2	1	58322	1000	3.00E-01	-2.00E-02	-1.99E-02	10.27
2	2	14762	784	3.00E-01	-2.44E-02	-2.26E-02	8.69
2	3	6642	686	3.00E-01	-2.56E-02	-2.27E-02	9.89
2	4	3782	915	3.00E-01	-2.66E-02	-3.80E-03	8.70
2	5	2450	1000	3.00E-01	-2.76E-02	-1.85E-02	10.67
2	6	1722	1000	2.98E-01	-1.59E-02	-1.30E-02	12.95
2	10	650	378	2.99E-01	-2.74E-02	-6.40E-04	15.44
3	2	32942	1000	3.00E-01	-2.33E-02	-2.33E-02	11.19
3	3	14762	1000	3.00E-01	-2.45E-02	-2.33E-02	11.03
3	4	8372	793	3.00E-01	-2.48E-02	-2.28E-02	10.46
3	5	5402	516	3.00E-01	-2.51E-02	-2.28E-02	10.47
3	6	3782	854	3.00E-01	-2.54E-02	-2.25E-02	10.05
3	10	1406	1000	2.96E-01	-5.03E-02	-5.23E-03	17.33
4	2	58322	888	3.00E-01	-2.35E-02	-2.30E-02	9.15
4	3	26082	1000	3.00E-01	-1.75E-02	-1.32E-02	9.87
4	4	14762	1000	3.00E-01	-2.42E-02	-2.31E-02	8.31
4	5	9506	843	3.00E-01	-2.45E-02	-2.32E-02	8.18
4	6	6642	564	3.00E-01	-2.46E-02	-2.28E-02	9.32
4	10	2450	954	3.00E-01	-2.56E-02	-1.81E-07	9.95
5	2	90902	896	3.00E-01	-2.39E-02	-2.28E-02	9.28
5	3	40602	901	3.00E-01	-2.40E-02	-2.35E-02	8.74
5	4	22952	821	3.00E-01	-2.46E-02	-2.30E-02	10.11
5	5	14762	1000	3.00E-01	-2.17E-02	-2.13E-02	7.65
5	6	10302	1000	2.99E-01	-2.21E-02	-2.11E-02	9.33
5	10	3782	764	3.00E-01	-2.48E-02	-2.30E-02	10.26

Table C.2: Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and sensitivity filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	26082	1000	3.00E-01	-2.46E-02	-2.23E-02	11.18
1	2	6642	1000	3.00E-01	-2.62E-02	-2.27E-07	6.11
1	4	1722	1000	2.84E-01	-1.97E-02	-8.08E-03	18.02
1	5	1122	1000	2.99E-01	-1.98E-02	-1.66E-02	10.30
1	8	462	1000	2.96E-01	1.96E-05	4.71E-09	12.04
1	10	306	1000	2.85E-01	7.18E-09	4.06E-10	20.46
2	1	103362	1000	3.00E-01	-2.46E-02	-2.43E-02	6.41
2	2	26082	1000	3.00E-01	-2.43E-02	-2.31E-02	9.41
2	4	6642	1000	3.00E-01	-2.68E-02	-2.07E-03	7.84
2	5	4290	958	3.00E-01	-2.70E-02	-2.32E-07	7.06
2	8	1722	1000	2.99E-01	-2.06E-02	-2.18E-05	12.85
2	10	1122	1000	2.99E-01	-9.13E-03	-6.88E-04	18.02
3	2	58322	952	3.00E-01	-2.49E-02	-2.45E-02	6.97
3	4	14762	747	3.00E-01	-2.56E-02	-2.44E-02	7.60
3	5	9506	843	3.00E-01	-2.52E-02	-2.24E-07	6.50
3	8	3782	909	3.00E-01	-2.61E-02	-2.10E-07	9.01
3	10	2450	677	3.00E-01	-2.83E-02	-1.97E-02	7.37
4	2	103362	662	3.00E-01	-1.97E-02	-1.95E-02	12.89
4	4	26082	1000	3.00E-01	-2.14E-02	-2.12E-02	7.96
4	5	16770	1000	3.00E-01	-2.47E-02	-2.41E-02	7.17
4	8	6642	1000	3.00E-01	-2.52E-02	-2.35E-02	7.02
4	10	4290	973	3.00E-01	-2.56E-02	-2.34E-02	8.26
5	2	161202	1000	3.00E-01	-2.38E-02	-2.40E-02	6.21
5	4	40602	925	3.00E-01	-2.45E-02	-2.45E-02	7.12
5	5	26082	1000	3.00E-01	-2.47E-02	-2.45E-02	6.98
5	8	10302	844	3.00E-01	-2.40E-02	-2.30E-02	8.15
5	10	6642	953	3.00E-01	-2.54E-02	-2.43E-02	7.64

Table C.3: Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	14762	1000	3.00E-01	-2.47E-02	-2.19E-02	10.99
1	2	3782	1000	3.00E-01	-2.58E-02	-2.14E-02	12.22
1	3	1722	1000	2.96E-01	-1.90E-02	-7.62E-03	17.10
1	4	992	1000	3.00E-01	-2.48E-02	-1.76E-02	11.22
1	5	650	1000	3.00E-01	-1.10E-01	-6.89E-08	14.65
1	6	462	1000	2.99E-01	1.64E-06	4.26E-09	18.77
1	10	182	397	3.00E-01	-5.91E-03	1.35E-11	12.85
2	1	58322	905	3.00E-01	-2.03E-02	-1.99E-02	12.56
2	2	14762	1000	3.00E-01	-2.38E-02	-2.21E-02	11.78
2	3	6642	1000	3.00E-01	-2.38E-02	-2.18E-02	14.09
2	4	3782	1000	3.00E-01	-2.36E-02	-2.11E-02	14.49
2	5	2450	1000	3.00E-01	-2.47E-02	-2.15E-02	12.20
2	6	1722	1000	3.00E-01	-2.48E-02	-1.96E-02	14.65
2	10	650	485	2.99E-01	-2.14E-02	-1.52E-02	18.20
3	2	32942	1000	3.00E-01	-1.87E-02	-1.87E-02	17.65
3	3	14762	1000	3.00E-01	-2.31E-02	-2.18E-02	12.94
3	4	8372	1000	3.00E-01	-2.32E-02	-2.16E-02	12.61
3	5	5402	1000	3.00E-01	-2.40E-02	-2.22E-02	11.39
3	6	3782	1000	3.00E-01	-2.37E-02	-2.18E-02	11.95
3	10	1406	1000	3.00E-01	-2.33E-02	-1.31E-02	14.47
4	2	58322	1000	3.00E-01	-2.25E-02	-2.20E-02	10.46
4	3	26082	587	3.00E-01	-1.75E-02	-1.74E-02	20.61
4	4	14762	1000	3.00E-01	-2.29E-02	-2.19E-02	10.72
4	5	9506	1000	3.00E-01	-2.33E-02	-2.21E-02	10.50
4	6	6642	1000	3.00E-01	-2.32E-02	-2.18E-02	10.26
4	10	2450	1000	3.00E-01	-2.37E-02	-2.15E-02	12.73
5	2	90902	1000	3.00E-01	-2.30E-02	-2.19E-02	10.56
5	3	40602	1000	3.00E-01	-2.24E-02	-2.19E-02	10.41
5	4	22952	1000	3.00E-01	-2.32E-02	-2.18E-02	10.65
5	5	14762	1000	3.00E-01	-2.21E-02	-2.14E-02	12.70
5	6	10302	1000	3.00E-01	-2.29E-02	-2.19E-02	10.51
5	10	3782	1000	3.00E-01	-2.35E-02	-2.20E-02	12.36

Table C.4: Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	26082	1000	3.00E-01	-2.41E-02	-2.25E-02	12.48
1	2	6642	1000	3.00E-01	-2.60E-02	-2.25E-02	10.99
1	4	1722	1000	3.00E-01	-3.01E-02	-1.79E-02	10.22
1	5	1122	1000	3.00E-01	-1.54E-02	-9.82E-03	14.76
1	8	462	1000	2.90E-01	-3.22E-06	-2.15E-09	19.28
1	10	306	1000	2.50E-01	-2.14E-05	-9.40E-09	23.15
2	1	103362	1000	3.00E-01	-2.32E-02	-2.27E-02	11.52
2	2	26082	1000	3.00E-01	-2.44E-02	-2.33E-02	10.87
2	4	6642	1000	3.00E-01	-2.46E-02	-2.27E-02	10.63
2	5	4290	1000	3.00E-01	-2.55E-02	-2.29E-02	10.07
2	8	1722	1000	3.00E-01	-2.67E-02	-2.15E-02	9.19
2	10	1122	1000	2.99E-01	-5.61E-02	-1.27E-02	10.25
3	2	58322	1000	3.00E-01	-2.30E-02	-2.24E-02	12.16
3	4	14762	1000	3.00E-01	-2.43E-02	-2.33E-02	9.09
3	5	9506	1000	3.00E-01	-2.41E-02	-2.28E-02	11.14
3	8	3782	1000	3.00E-01	-2.49E-02	-2.29E-02	9.60
3	10	2450	1000	3.00E-01	-2.49E-02	-2.14E-02	9.55
4	2	103362	207	3.00E-01	-1.86E-02	-1.82E-02	23.91
4	4	26082	786	3.00E-01	-1.90E-02	-1.89E-02	19.78
4	5	16770	1000	3.00E-01	-2.19E-02	-2.14E-02	8.88
4	8	6642	1000	3.00E-01	-2.24E-02	-2.15E-02	11.96
4	10	4290	1000	3.00E-01	-2.42E-02	-2.26E-02	10.35
5	2	161202	1000	3.00E-01	-2.24E-02	-2.25E-02	11.31
5	4	40602	1000	3.00E-01	-2.30E-02	-2.27E-02	12.47
5	5	26082	1000	3.00E-01	-2.36E-02	-2.30E-02	9.73
5	8	10302	1000	3.00E-01	-2.34E-02	-2.28E-02	10.22
5	10	6642	1000	3.00E-01	-2.38E-02	-2.29E-02	8.61

Table C.5: Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	14762	463	3.00E-01	-2.49E-02	-2.12E-02	0.72
1	2	3782	398	3.00E-01	-2.84E-02	-2.25E-02	0.80
1	3	1722	1000	2.96E-01	-2.64E-02	-1.52E-02	7.35
1	4	992	607	3.00E-01	-3.57E-02	-1.39E-06	0.41
1	5	650	445	3.00E-01	-6.05E-02	-4.28E-09	0.22
1	6	462	348	2.95E-01	-6.28E-04	8.47E-13	9.55
1	10	182	384	3.00E-01	-7.45E-03	7.23E-11	0.87
2	1	58322	353	2.98E-01	-1.72E-02	-1.76E-02	5.99
2	2	14762	367	3.00E-01	-2.46E-02	-2.28E-02	0.66
2	3	6642	366	3.00E-01	-2.55E-02	-2.26E-02	0.63
2	4	3782	356	3.00E-01	-2.56E-02	-2.18E-02	0.59
2	5	2450	328	3.00E-01	-2.57E-02	-6.28E-07	1.05
2	6	1722	400	3.00E-01	-2.83E-02	-1.27E-02	0.37
2	10	650	1000	2.98E-01	-2.79E-02	-1.05E-05	5.52
3	2	32942	1000	2.99E-01	-2.25E-02	-2.38E-02	4.22
3	3	14762	418	3.00E-01	-2.41E-02	-2.28E-02	1.03
3	4	8372	325	3.00E-01	-2.45E-02	-2.25E-02	0.49
3	5	5402	380	3.00E-01	-2.50E-02	-2.28E-02	0.60
3	6	3782	373	3.00E-01	-2.42E-02	-5.10E-07	0.72
3	10	1406	378	3.00E-01	-2.55E-02	-1.47E-06	1.25
4	2	58322	393	3.00E-01	-2.32E-02	-2.26E-02	0.71
4	3	26082	450	3.00E-01	-2.27E-02	-2.19E-02	0.95
4	4	14762	426	3.00E-01	-2.40E-02	-2.28E-02	0.86
4	5	9506	432	3.00E-01	-2.41E-02	-1.69E-08	1.01
4	6	6642	422	3.00E-01	-2.40E-02	-2.13E-02	0.84
4	10	2450	383	3.00E-01	-2.40E-02	-2.15E-06	1.57
5	2	90902	372	3.00E-01	-2.25E-02	-2.16E-02	0.82
5	3	40602	515	3.00E-01	-2.22E-02	-2.17E-02	0.83
5	4	22952	434	3.00E-01	-2.41E-02	-2.24E-02	0.98
5	5	14762	374	3.00E-01	-2.12E-02	5.40E-07	3.24
5	6	10302	390	3.00E-01	-2.34E-02	-2.02E-06	0.85
5	10	3782	378	3.00E-01	-2.35E-02	-1.46E-02	1.59

Table C.6: Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering with Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	26082	418	3.00E-01	-2.48E-02	-2.27E-02	0.97
1	2	6642	415	3.00E-01	-2.71E-02	-2.26E-02	0.86
1	4	1722	392	3.00E-01	-2.35E-02	-6.51E-03	1.00
1	5	1122	674	2.99E-01	-1.97E-02	-1.11E-02	6.15
1	8	462	832	2.80E-01	6.63E-03	1.51E-09	17.75
1	10	306	385	2.93E-01	-9.11E-03	1.19E-09	13.80
2	1	103362	448	3.00E-01	-2.30E-02	-2.22E-02	1.00
2	2	26082	687	3.00E-01	-2.43E-02	-2.29E-02	1.16
2	4	6642	464	3.00E-01	-2.48E-02	-2.21E-06	0.74
2	5	4290	398	3.00E-01	-2.54E-02	-1.97E-02	0.86
2	8	1722	356	3.00E-01	-2.73E-02	-3.27E-07	0.80
2	10	1122	387	3.00E-01	-3.03E-02	-3.98E-07	0.83
3	2	58322	533	3.00E-01	-2.25E-02	-2.21E-02	1.52
3	4	14762	463	3.00E-01	-2.36E-02	-2.28E-02	0.87
3	5	9506	405	3.00E-01	-2.42E-02	-1.46E-06	0.68
3	8	3782	461	3.00E-01	-2.54E-02	-4.51E-08	1.02
3	10	2450	363	3.00E-01	-2.57E-02	-1.28E-05	1.08
4	2	103362	402	3.00E-01	-2.26E-02	-2.37E-02	0.92
4	4	26082	1000	3.00E-01	-2.37E-02	-2.33E-02	1.88
4	5	16770	482	3.00E-01	-2.28E-02	-3.77E-04	0.87
4	8	6642	403	3.00E-01	-2.39E-02	-1.17E-02	1.01
4	10	4290	362	3.00E-01	-2.37E-02	-2.18E-02	0.94
5	2	161202	382	3.00E-01	-2.33E-02	-2.35E-02	1.23
5	4	40602	464	3.00E-01	-2.16E-02	-2.15E-02	1.06
5	5	26082	430	3.00E-01	-2.46E-02	-1.84E-02	1.06
5	8	10302	359	3.00E-01	-2.37E-02	-2.30E-02	0.90
5	10	6642	350	3.00E-01	-2.37E-02	-2.07E-02	0.96

Table C.7: Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	14762	306	2.98E-01	-1.99E-02	-1.83E-02	0.22
1	2	3782	305	2.98E-01	-2.16E-02	-1.88E-02	0.19
1	3	1722	306	2.99E-01	-2.49E-02	-1.96E-02	0.18
1	4	992	318	2.88E-01	-1.86E-02	-8.08E-09	1.67
1	5	650	315	2.80E-01	-1.74E-02	7.03E-09	2.83
1	6	462	351	2.99E-01	9.14E-02	-6.63E-03	0.09
1	10	182	308	2.99E-01	-7.50E-03	6.85E-12	0.21
2	1	58322	375	2.94E-01	-1.67E-02	-1.82E-02	4.53
2	2	14762	382	3.00E-01	-2.32E-02	-2.15E-02	0.25
2	3	6642	305	2.98E-01	-2.19E-02	-2.02E-02	0.19
2	4	3782	314	2.99E-01	-2.14E-02	-1.21E-02	0.28
2	5	2450	389	3.00E-01	-4.35E-02	-3.04E-03	0.41
2	6	1722	326	2.34E-01	-2.20E-02	-1.21E-02	10.34
2	10	650	333	2.86E-01	-2.09E-02	-1.17E-02	2.80
3	2	32942	327	2.92E-01	-1.81E-02	-1.83E-02	2.85
3	3	14762	428	3.00E-01	-2.07E-02	-1.98E-02	0.55
3	4	8372	351	2.98E-01	-2.30E-02	-2.13E-02	0.26
3	5	5402	329	2.97E-01	-2.29E-02	-2.13E-02	0.39
3	6	3782	305	2.98E-01	-2.32E-02	-2.10E-02	0.20
3	10	1406	486	2.99E-01	-2.20E-02	-1.55E-02	0.90
4	2	58322	398	3.00E-01	-1.92E-02	-1.88E-02	0.45
4	3	26082	353	2.88E-01	-1.50E-02	-1.53E-02	4.92
4	4	14762	308	3.00E-01	-2.07E-02	-2.00E-02	0.43
4	5	9506	418	3.00E-01	-2.12E-02	-1.98E-02	0.39
4	6	6642	336	2.98E-01	-2.32E-02	-2.09E-02	0.21
4	10	2450	320	2.98E-01	-2.27E-02	-2.07E-02	0.30
5	2	90902	317	3.00E-01	-2.22E-02	-2.13E-02	0.31
5	3	40602	348	3.00E-01	-2.15E-02	-2.09E-02	0.40
5	4	22952	372	3.00E-01	-2.28E-02	-2.09E-02	0.26
5	5	14762	330	2.83E-01	-1.84E-02	-1.77E-02	5.50
5	6	10302	311	2.99E-01	-2.09E-02	-2.00E-02	0.20
5	10	3782	349	2.98E-01	-2.35E-02	-2.14E-02	0.26

Table C.8: Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and density filtering with modified Heaviside projection is applied with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	26082	453	3.00E-01	-2.14E-02	-1.98E-02	0.42
1	2	6642	301	2.98E-01	-2.51E-02	-2.19E-02	0.21
1	4	1722	306	2.99E-01	-2.24E-02	-1.83E-02	0.22
1	5	1122	326	2.90E-01	-1.63E-02	-1.33E-02	1.74
1	8	462	372	3.00E-01	-7.58E-02	-1.45E-08	0.11
1	10	306	302	1.78E-01	2.89E-09	5.04E-10	4.77
2	1	103362	388	3.00E-01	-2.15E-02	-2.10E-02	0.47
2	2	26082	323	3.00E-01	-2.28E-02	-2.21E-02	0.24
2	4	6642	453	3.00E-01	-2.25E-02	-2.11E-02	0.35
2	5	4290	307	2.98E-01	-2.32E-02	-1.94E-02	0.23
2	8	1722	331	2.99E-01	-2.53E-02	-5.54E-08	0.22
2	10	1122	430	3.00E-01	-2.60E-02	-4.64E-08	0.33
3	2	58322	340	3.00E-01	-2.22E-02	-2.17E-02	0.41
3	4	14762	428	3.00E-01	-1.86E-02	-1.79E-02	0.37
3	5	9506	401	3.00E-01	-2.45E-02	-2.29E-02	0.35
3	8	3782	320	2.97E-01	-2.22E-02	-2.02E-02	0.41
3	10	2450	315	2.98E-01	-2.47E-02	-2.10E-02	0.45
4	2	103362	403	2.96E-01	-1.71E-02	-1.69E-02	2.03
4	4	26082	355	2.82E-01	-1.56E-02	-1.52E-02	5.15
4	5	16770	496	2.97E-01	-1.74E-02	-1.69E-02	1.61
4	8	6642	334	2.98E-01	-2.31E-02	-2.12E-02	0.26
4	10	4290	327	2.98E-01	-2.32E-02	-2.06E-02	0.31
5	2	161202	462	3.00E-01	-2.10E-02	-2.11E-02	0.53
5	4	40602	393	3.00E-01	-2.22E-02	-2.19E-02	0.33
5	5	26082	371	3.00E-01	-1.99E-02	-1.93E-02	0.43
5	8	10302	355	2.94E-01	-1.84E-02	-1.74E-02	2.76
5	10	6642	478	3.00E-01	-2.22E-02	-2.12E-02	0.41

Table C.9: Results of the stability experiments on the compliant force inverter example. A discretization of 120 by 60 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	14762	386	2.99E-01	-2.08E-02	-1.95E-02	0.73
1	2	3782	383	2.97E-01	-1.94E-02	-1.81E-02	1.03
1	3	1722	538	2.99E-01	-1.63E-02	-2.98E-09	0.95
1	4	992	362	3.00E-01	-1.60E-02	-1.11E-08	1.09
1	5	650	555	3.05E-01	-1.98E-02	4.17E-13	0.92
1	6	462	369	2.96E-01	3.77E-02	-8.75E-10	0.64
1	10	182	478	3.01E-01	-6.29E-03	1.17E-10	1.00
2	1	58322	304	2.99E-01	-1.63E-02	-1.60E-02	0.49
2	2	14762	405	3.01E-01	-2.06E-02	-1.97E-02	1.01
2	3	6642	459	2.99E-01	-2.18E-02	-2.03E-02	1.13
2	4	3782	393	3.00E-01	-2.11E-02	-2.00E-02	0.86
2	5	2450	497	2.98E-01	-1.99E-02	-1.87E-02	1.14
2	6	1722	507	3.01E-01	-1.96E-02	-1.79E-02	0.98
2	10	650	466	3.00E-01	-1.93E-02	-1.42E-02	0.85
3	2	32942	255	3.01E-01	-1.66E-02	-1.63E-02	0.60
3	3	14762	475	2.98E-01	-1.66E-02	-1.62E-02	0.77
3	4	8372	433	2.98E-01	-1.93E-02	-1.88E-02	0.97
3	5	5402	379	2.99E-01	-2.22E-02	-2.13E-02	0.82
3	6	3782	399	3.00E-01	-2.03E-02	-1.93E-02	0.92
3	10	1406	414	3.00E-01	-2.08E-02	-1.82E-02	1.12
4	2	58322	372	3.00E-01	-1.82E-02	-1.79E-02	0.70
4	3	26082	304	2.99E-01	-1.62E-02	-1.70E-02	0.71
4	4	14762	391	3.00E-01	-1.81E-02	-1.78E-02	0.47
4	5	9506	399	2.99E-01	-1.98E-02	-1.92E-02	0.83
4	6	6642	379	2.99E-01	-1.41E-02	-1.39E-02	0.60
4	10	2450	372	2.99E-01	-1.90E-02	-1.83E-02	0.93
5	2	90902	356	3.00E-01	-1.95E-02	-1.88E-02	0.62
5	3	40602	381	3.00E-01	-1.93E-02	-1.90E-02	0.65
5	4	22952	362	3.00E-01	-1.92E-02	-1.81E-02	0.89
5	5	14762	353	2.99E-01	-1.88E-02	-1.85E-02	0.47
5	6	10302	367	3.00E-01	-1.79E-02	-1.78E-02	0.88
5	10	3782	400	2.98E-01	-1.89E-02	-1.85E-02	0.77

Table C.10: Results of the stability experiments on the compliant force inverter example. A discretization of 160 by 80 voxels is used, and robust topology optimization is applied ($\eta = 0.2$) with $r_{min} = 2$ voxel widths

p [-]	n_{voxel} [-]	n_{dof} [-]	n_{iter} [-]	V	$\hat{F}(\boldsymbol{\rho})$ [-]	u_{out} [m]	M_{nd} [%]
1	1	26082	398	3.00E-01	-2.15E-02	-2.06E-02	0.79
1	2	6642	405	3.01E-01	-1.57E-02	-1.51E-02	0.65
1	4	1722	447	2.99E-01	-1.56E-02	9.83E-09	0.14
1	5	1122	647	3.00E-01	-1.42E-02	-6.40E-09	0.51
1	8	462	937	2.62E-01	-8.37E-09	7.22E-10	1.62
1	10	306	398	3.01E-01	-1.63E-02	9.39E-10	0.44
2	1	103362	359	3.00E-01	-1.41E-02	-1.36E-02	0.43
2	2	26082	378	3.01E-01	-1.23E-02	-1.23E-02	0.48
2	4	6642	366	3.00E-01	-2.20E-02	-2.04E-02	0.79
2	5	4290	429	2.99E-01	-1.81E-02	-1.75E-02	0.65
2	8	1722	490	2.92E-01	-1.71E-02	1.63E-08	0.54
2	10	1122	366	3.00E-01	-1.95E-02	-1.30E-02	0.13
3	2	58322	387	3.00E-01	-1.38E-02	-1.35E-02	0.34
3	4	14762	393	3.00E-01	-1.86E-02	-1.84E-02	0.58
3	5	9506	399	3.00E-01	-1.34E-02	3.28E-04	0.52
3	8	3782	506	2.99E-01	-2.20E-02	-2.10E-02	0.96
3	10	2450	374	2.98E-01	-1.76E-02	-1.68E-02	0.57
4	2	103362	304	2.99E-01	-1.25E-02	-1.28E-02	0.35
4	4	26082	205	1.92E-01	1.11E-09	5.63E-10	1.62
4	5	16770	348	3.00E-01	-1.45E-02	-1.46E-02	0.37
4	8	6642	373	2.99E-01	-1.64E-02	-1.62E-02	0.69
4	10	4290	413	2.99E-01	-2.14E-02	-1.94E-02	0.90
5	2	161202	370	3.00E-01	-1.37E-02	-1.38E-02	0.53
5	4	40602	393	3.01E-01	-1.41E-02	-1.41E-02	0.54
5	5	26082	397	3.00E-01	-1.40E-02	-1.36E-02	0.62
5	8	10302	477	2.99E-01	-1.77E-02	-1.78E-02	0.90
5	10	6642	665	3.00E-01	-1.79E-02	-1.76E-02	0.63

